

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Konfigurovatelná aplikace hodinek pro platformu Garmin
Connect IQ**

**Configurable Watch Application for Garmin Connect IQ
Platform**

2019

Tomáš Věšek

Zadání bakalářské práce

Student:

Tomáš Věšek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Konfigurovatelná aplikace hodinek pro platformu Garmin Connect IQ
Configurable Watch Application for Garmin Connect IQ Platform

Jazyk vypracování:

čeština

Zásady pro vypracování:

Garmin Connect IQ je jednou z platform určených pro zařízení jako jsou chytré nebo sportovní hodinky. Většina aplikací zobrazujících čas a data ze senzorů zařízení není konfigurovatelná, cílem této práce je navrhnout a naimplementovat řadu komponent uživatelského rozhraní (UI) ze kterých si bude moci uživatel sestavit vlastní aplikaci.

1. Nastudujte platformu Garmin Connect IQ.
2. Navrhněte komponenty UI zobrazující čas a data ze senzorů zařízení (stav baterie, srdeční tep, počet kroků, atd.).
3. Napište aplikaci umožňující sestavit z těchto komponent vlastní aplikaci pro zařízení.
4. Otestujte aplikaci na nejméně dvou zařízeních s různým rozlišením displeje.
5. Porovnejte implementovanou aplikaci s existujícími aplikacemi pro tuto platformu.

Seznam doporučené odborné literatury:

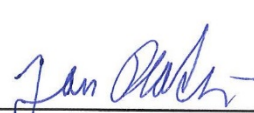
Dle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **doc. Ing. Michal Krátký, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



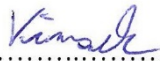
prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty



Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 28. dubna 2019


.....
podpis studenta

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Michalovi Krátkému, Ph.D. za jeho odbornou pomoc, cenné rady a konzultace poskytnuté při vytváření této bakalářské práce.

Abstrakt

Práce se zabývá vývojem aplikací pro zařízení Garmin. Dále platformou pro programování aplikací na nositelná zařízení Connect IQ a objektově orientovaným jazykem Monkey C. V práci je popsáno konkrétní řešení vývoje aplikace pro modely hodinek Garmin Forerunner 235 (dále fr235) a Garmin vívoActive HR. Tento typ hodinek spadá do kategorie sportovních hodinek, které slouží zejména k zaznamenání fyzických aktivit a kromě standardního zobrazování času disponují mnoha dalšími funkcemi popsány v práci. Na modelu fr235 je otestována funkčnost pro zařízení obsahující pouze fyzická tlačítka a mající kulatý displej, zatímco u modelu vívoActive HR je funkčnost otestována na hranatém dotykovém displeji. Cílem této práce je vytvořit konfigurovatelnou aplikaci tří typů widget, vzhled hodinek a samotný typ aplikace. Každý uživatel si tak může sestavit svou vlastní aplikaci dle vlastních požadavků, omezenou pouze možnostmi daného zařízení. Aktuální trh s aplikacemi stejného typu tuto konfigurovatelnost nenabízí.

Klíčová slova

Garmin; Monkey C; konfigurovatelné aplikace; platforma Connect IQ; Garmin Forerunner 235; Garmin vívoActive HR;

Abstract

This bachelor thesis deals with the development of applications for Garmin devices. Continuous with platform for programming applications on a Connect IQ wearable device, and an object-oriented language Monkey C. This thesis describes the application development for models Garmin Forerunner 235 (shortcut fr235) and Garmin vívoActive HR. These types of watch fall into the category of sports watches, which are mainly used to record physical activities and, in addition of the standard display of time, has many other functions described in this work. Fr235 is tested for functionality with only physical buttons and a round display, while the vívoActive HR model uses a square touch screen. The aim of this work is to create a three types of configurable application widget, watchface and application itself. Each user can build their own application according to further requirements, limited only by the capabilities of the device. The current market of downloadable applications of the same type does not offer such an application.

Key words

Garmin; Monkey C; configurable applications; Connect IQ platform; Garmin Forerunner 235; Garmin vívoActive HR;

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
G	m/s	přetížení

Seznam použitých zkratek

Zkratka	Význam
GPS	Global Positioning System
PDA	Personal Digital Assistant
HR	Heart Rate
px	pixel
API	Application Programming Interface
iOS	iPhone OS
XML	eXtensible Markup Language
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
UUID	Universally Unique Identifier
IDE	Integrated Development Environment
URI	Uniform Resource Identifier
ISO	International Organization for Standardization
SDK	Software development kit
UI	User interface
ATM	Atmospheric pressure
Fr	Forerunner
USB	Universal Serial Bus

Obsah

1	Sportovní hodinky	- 12 -
1.1	Garmin.....	- 12 -
1.1.1	Forerunner	- 14 -
1.1.2	VívoActive	- 15 -
2	Vývoj aplikací pro chytré hodinky	- 17 -
3	Monkey C.....	- 19 -
3.1	Překladač	- 20 -
3.2	Datové úložiště	- 21 -
4	Platforma Connect IQ.....	- 23 -
4.1	Konfigurační soubor.....	- 26 -
4.2	Kvalifikátory	- 27 -
4.3	API pro práci s uživatelským rozhraním	- 28 -
4.4	Vykreslovací kontext.....	- 29 -
4.5	Manipulace se vstupy	- 29 -
4.5.1	Nabídka Menu	- 30 -
4.5.2	Generický a číselný výběr	- 31 -
4.5.3	Potvrzovací dialog a dialog průběhu	- 32 -
4.6	Pozicování a senzory	- 32 -
5	Specifikace zadání.....	- 33 -
6	Analýza a návrh.....	- 35 -
6.1	Funkční požadavky	- 35 -
6.2	Seznam podporovaných metrik	- 35 -
6.3	Stavová analýza aplikace.....	- 36 -
6.4	Návrh uživatelského rozhraní.....	- 37 -
7	Implementace konfigurovatelné aplikace.....	- 39 -
7.1	Vytvoření projektu	- 39 -
7.2	Zvolené nástroje	- 39 -
7.3	Definování problémů aplikace	- 41 -
7.4	Popis implementace.....	- 41 -

8	Testování	- 46 -
8.1	Směr vývoje.....	- 46 -
8.2	Ukázky a srovnání aplikace.....	- 47 -
	Závěr	- 52 -
	Použitá literatura	lii

Úvod

Tato bakalářská práce se zabývá vývojem konfigurovatelných aplikací na platformě Garmin Connect IQ [12] pro zařízení Garmin Forerunner 235 (dále fr235) a Garmin vívoActive HR (dále jen vívoActive HR). Popisuje také nositelná zařízení, konkrétně chytré hodinky od této společnosti. Hlavními tématy teoretické části práce je obecné téma sportovních hodinek a jejich parametry, historický vývoj společnosti Garmin a využití hodinek v každodenním životě. V kapitole 1 je popsáno jak se sportovní hodinky liší od klasických a proč byl v práci vybrán právě Garmin, který je následně představen v první podkapitole. Po teoretickém úvodu a seznámení s firmou Garmin budou popsány odlišnosti dvou konkrétních modelů, tlačítkového modelu fr235 a dotykového modelu vívoActive HR. Dále práce popisuje vývoj aplikací pro chytré hodinky a rozdíl oproti vývoji aplikací na chytrých telefonech, které na trh přicházely současně. V práci jsou také specifikovány výhody chytrých hodinek oproti chytrým telefonům a srovnání modelů předních dodavatelů. V kapitole 3 je popsán programovací jazyk a funkcionality překladače Monkey C. V následující kapitole 4 je uvedeno seznámení s platformou Garmin Connect IQ. Na první část by bylo vhodné nahlížet jako na zdroj informací k vytvoření představy o této problematice.

Praktická část této práce začínající kapitolou 5, popisuje použité nástroje, vývoj aplikace, analýzu aplikace a potřebné nastavení vývojové prostředí Eclipse, nezbytné pro sestavení aplikace. V kapitole 6 se práce zabývá konkrétním popisem návrhu, seznámení s konfigurovatelnou aplikací, funkcionalitou aplikace, informacemi týkající se testování, definování zjištěných problémů, jejich zpracování a samotnou implementací vytvořené konfigurovatelné aplikace. V praktické části práce jsem také uvedl návrhy na další směr vývoje aplikace a srovnání s již existujícími aplikacemi, které jsou konfigurovatelné aplikaci nejvíce podobné.

1 Sportovní hodinky

Sportovní hodinky nebo-li sporttestery přišly na trh již před více než 40 lety a to zásluhou firmy Polar [1]. Od klasických hodinek se liší především tím, že uživateli nabízí, kromě zobrazení času, mnoho dalších funkcí. Mezi základní funkce patří stopky, měření vzdálenosti běhu či chůze, počet spálených kalorií či krokoměr. V uživatelském nastavení lze zvolit pohlaví, váhu, výšku a věk uživatele. Sportovní hodinky mají možnost komunikovat s dalšími zařízeními pomocí technologie Bluetooth [46] nebo ANT+ [35]. V nejnovějších modelech jako například v Garmin Fenix 5 [4] jsou zabudovány funkce odesílání sms zpráv, přijímání hovorů (pouze však v případě, že se nachází mobilní telefon v blízkosti signálu Bluetooth), navigace (určování polohy pomocí GPS signálu), připojení k Internetu prostřednictvím Wi-Fi [47], bezkontaktní platby, měření tepové frekvence a další. Existují dva způsoby měření tepové frekvence a to pomocí elektrického potenciálu způsobeného srdeční aktivitou, jako je tomu u hrudních pásů (přesnější naměřené hodnoty), nebo měření pomocí optického čidla a proudění krve skrze pokožku (pohodlnější). Výrobci experimentují s různými barvami diod, které prosvětlují pokožku, zatím však je u všech modelů barva diody zelená. Mezi hlavní výhody hodinek patří jednoduchá ovladatelnost, intuitivní prostředí, nízká hmotnost a odolnost. Většina hodinek disponuje tvrzeným sklem odolným proti nárazům, odření či tlaku vznikajícím při sportovních aktivitách. Nevýhodou zařízení je slabá výdrž baterie, která se pohybuje v jednotkách hodin, pokud uživatel při své aktivitě zaznamenává měření srdečního tepu a zároveň využívá GPS. Rozhodujícími faktory týkající se výdrže baterie jsou značka a model. Stejně jako u mobilních telefonů běžících na OS Android [10], iOS [64] a dalších, můžeme na malých obrazovkách chytrých hodinek vidět ikony aplikací, jež je možné spouštět a lze hrát dokonce i jednoduché hry. Hlavní účel chytrých hodinek, ale nalezneme spíše v propojení s našim mobilním telefonem. Na hodinkách si tak můžeme číst zprávy, přijímat hovory, platit v bankách, dostávat upozornění (notifikace), zadávat hlasové povely a další.

Mezi tři nejvýznamnější výrobce sportovních hodinek patří zejména již zmíněná firma Polar, finská společnost Suunto [2] a americká společnost Garmin [6]. Poslední jmenovaná je z našeho pohledu nejzajímavější, jelikož byla platforma Garmin Connect IQ vybrána pro vypracování konfigurovatelné aplikace. Společnost Garmin se na sportovní hodinky nezaměřovala od svého počátku, navzdory tomu však získala velký počet zákazníků, zejména vzhledem svým modelům a cenovou dostupností. Popularitu lze vysledovat i z ročních obrátů firmy. V roce 1995 společnost uvedla zisk ve výši 105 miliónů amerických dolarů a o 4 roky později činil zisk již 233 miliónů dolarů. Nejaktuálnější data [7] z konce roku 2018 uvádí zisk 932 miliónů dolarů za svá zařízení. Firma Polar za uplynulý rok uvádí zisk 193 miliónů dolarů [48] a společnost Suunto 19 miliónů dolarů [49].

1.1 Garmin

Garmin Ltd. je firma, kterou v roce 1989 založili Gary Burrell a Min Kao. Ve svých počátcích se firma zaměřovala na vývoj leteckých, námořních, outdoorových a automobilových navigací využívajících GPS. Nejvýznamnější produkty, společně s daty uvedení na trh, můžete

nalézt v [5][6]. V roce 1991 přichází společnost s prvním GPS navigačním zařízením pojmenovaným GPS 100, které je na obrázku 1.1 ještě s původním jménem společnosti ProNav. O dva roky později vydal Garmin první přenosný přístroj s pohyblivou mapou nazvaný GPS 95.



Obrázek 1.1: *GPS 100 první navigační zařízení firmy Garmin*

V roce 1997 představil Garmin dva přelomové přístroje a to GPS 12, který svým vzhledem připomíná mobilní telefon, a GPS III, který má rozměry automobilového rádia. S expanzí mobilních telefonů, umožňujících přijímat signál GPS, přišel Garmin v roce 1998 s telefonem se zabudovanými mapovými podklady NavTalk. Ve stejném roce uvedli první přenosnou GPS navigaci pro automobily s názvem StreetPilot.

Další prvenství přinesl firmě rok 1999, kdy představila řadu eTrex. Byla to jediná ruční GPS navigace na trhu, která je dodnes cenově nejdostupnějším a nejprodávanějším produktem společnosti. V roce 2001 vyvinuli GPS s hlasovou navigací nazývaný se StreetPilot III a o rok později vytvořili Personal Digital Assistant (PDA)¹ s GPS iQue a také první vysílačku s modulem GPS Rino. Rok 2003 byl průlomový s nastupujícími chytrými hodinkami. Byly vydány řady Forerunner 101, 201 a 301, které firma vyvinula pro sportovní účely. Trojici řad doplnily dále řady vívoActive a Fenix. V kapitolách 1.1.1 a 1.1.2 budu detailně popisovat dvě řady a to Forerunner a vívoActive HR, na kterých budou probíhat testy konfigurovatelné aplikace. V roce 2005 přišla firma Garmin s revoluční automobilovou navigací, řadou nüvi a o rok později vydala řadu zūmo, která je určena pro motocykly. Řada má zabudovanou MP3, text-to-speech, Bluetooth a další funkce. V roce 2013 uvedla řadu echoMap, což je GPS plotter s displejem a vestavěným sonarem. Začátkem roku 2016 představila firma sérii Drive určenou pro automobily, která

¹ PDA je malý kapesní počítač disponující dotykovou obrazovkou ovládanou perem (které se nazývá stylus).

obsahuje moderní softwarové řešení pro bezpečnější jízdu. Jednotlivé nejvýznamnější produkty společně s datem uvedení na trh lze nalézt v tabulce 1.1.

Tabulka 1.1: *Přehled produktů s datem uvedení na trh*

Název	Rok vydání
GPS 100	1991
GPS 95	1993
GPS 12, GPS III	1997
Navtalk, StreetPilot	1998
eTrex	1999
StreetPilot III	2001
GPS iQue, GPS Rino	2002
Forerunner, vívoActive, Fenix	2003
nüvi	2005
zūmo	2006
echoMap	2013
Drive	2016

Dále budou detailně rozebrány dvě modelové řady chytrých hodinek společnosti Garmin. Vybral jsem řadu Forerunner, protože sám vlastním zařízení fr235 a mohu ho tak jednoduše a rychle testovat. Jako druhou řadu jsem zvolil vívoActive, jelikož zde figuruje model vívoActive HR, který je diametrálně odlišný od všech ostatních modelů (zásluhou svého tvaru a rozlišení displeje) a je to tak ideální kandidát pro testování aplikací.

1.1.1 Forerunner

Forerunner je řada zaměřená především pro běžce. Lze ji však využít i pro cyklistiku a další sportovní aktivity. Novější modely této řady disponují optickým snímačem tepové frekvence na spodní straně hodinek, který kromě snímání tepové frekvence pomáhá i s odhadem spálených kalorií, analýzou spánku, stresu či případného celkového stavu organismu jedince. Zatímco první řady fr101, fr201, fr301 disponovaly obrazovkami ve tvaru obdélníku, nyní se nové modely vyrábějí pouze s kulatými displeji. Zároveň je dnes kladen velký důraz na elegantní design hodinek pro běžné nošení. Hodinky řady Forerunner disponují tlačítky, které jsou umístěny na boku displeje, dotykovým systémem či kombinací obojího. Rozlišení displeje u modelu Forerunner 235 znázorněného na obrázku 1.2 je 215 x 180px, velikost ikon 40 x 40px a disponuje

pouze 16 druhů barev. Další parametry lze najít v tabulce 1.2. Model 235 se ovládá pomocí pěti mechanických tlačítek, které ovládají funkce zapnutí/vypnutí, nahoru, dolů, potvrzení, zpět a menu. Tlačítka zařízení lze zamknout, aby se omylem neaktivovaly během aktivity nebo v režimu hodinek. Podrobné informace lze najít v [8].



Obrázek 1.2: *Forerunner 235*

Tabulka 1.2: *Vlastnosti a parametry modelu Forerunner 235*

Vlastnost/Parametr	Hodnota
Rozlišení	215 x 180px
Velikost ikon	40 x 40px
Počet barev	16
Výdrž baterie bez GPS	až 9 dní
Výdrž baterie s GPS	11 hodin
Hmotnost	54g
Voděodolnost	5 ATM
Odolnost	6G

1.1.2 **VívoActive**

Řada VívoActive patří mezi multisportovní hodinky, na které je možné zaznamenávat několik druhů aktivit. Již od prvních modelů této řady se hodinky ovládají pomocí dotyku. U modelu VívoActive 3 lze také ovládat hodinky pomocí speciální aktivní plochy na hraně

displeje, která nahrazuje tlačítka nahoru a dolů. Tento model je navíc symetrický a tak si může uživatel zvolit, na jaké straně bude mít jediné fyzické tlačítko. Hodinky je tedy možné pohodlně použít na levou i pravou ruku, v nastavení uživatel pak jen zvolí opačné otočení displeje. Model vivoActive HR znázorněný na obrázku 1.3 má rozlišení displeje 148 x 205px, velikost ikon 40 x 33px a disponuje 64 barvami. Další parametry lze najít v tabulce 1.3. Má pouze dvě fyzické tlačítka ovládající potvrzení a zpět. Podrobná specifikace je uvedena v [9].



Obrázek 1.3: *VivoActive HR*

Tabulka 1.3: *Vlastnosti a parametry modelu vivoActive HR*

Vlastnost/Parametr	Hodnota
Rozlišení	148 x 205px
Velikost ikon	40 x 33px
Počet barev	64
Výdrž baterie bez GPS	až 21 dní
Výdrž baterie s GPS	10 hodin
Hmotnost	38g
Voděodolnost	5 ATM
Odolnost	6G

2 Vývoj aplikací pro chytré hodinky

Operačních systémů (dále OS) pro vývoj aplikací pro chytré hodinky jsou na trhu desítky. Mezi ty největší patří Android Wear od společnosti Google [10], který byl představen v červnu roku 2014. Jedná se o operační systém podobný tomu, jaký nalezneme na našich chytrých telefonech (OS Android), s tím rozdílem, že je přizpůsobený na menší obrazovky. Wear OS, jak je nově tento operační systém nazýván, je postaven na programovacím jazyku Java. Ve svých zařízeních ho využívá mimo jiné i první firma, která přišla s chytrými hodinkami a to firma Polar. Wear OS podporuje dnes již standardní technologie Bluetooth, Wi-Fi, sítě 3G i LTE a další. Funkci, kterou tento systém jako první poskytl ve svých zařízeních, je automatické rozsvícení displeje při pohledu na obrazovku hodinek.

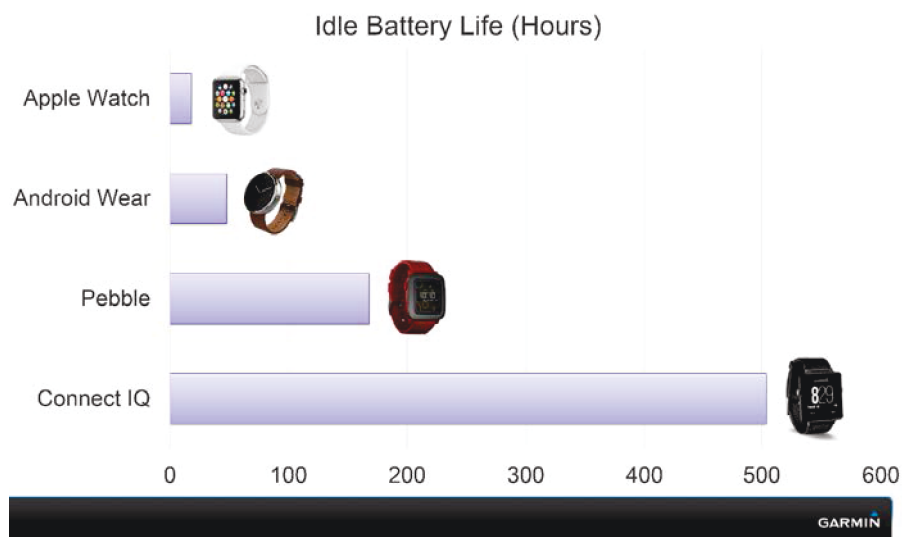
Finský zástupce na trhu, společnost Suunto [2], využívá vývojové prostředí nazvané Movesense [65]. Movesense používá objektově orientovaný jazyk C++ [66]. Movesense je vlastně balíček vývojářských nástrojů, API a samotného stejnojmenného snímače [36] potřebného k získání dat (počet kroků, rychlost a další) a vývoji specializovaných aplikací a služeb pro sportovní aktivity. Samotné API Movesense může komunikovat s mobilními telefony, které využívají iOS i Android, ale je nutné doinstalovat potřebné knihovny [11], což může být bráno jako nevýhoda. Naopak výhodou je samotný snímač Movesense (akcelerometr²), umístěný uvnitř zařízení Suunto, který se stará o výpočty dat zobrazovaných na hodinkách. Tento snímač je velice malý a s baterií váží pouhých 10g, navíc je konfigurovatelný. V zařízení Garmin Varia Vision [67] je pro srovnání použit akcelerometr, který váží 28g.

Na rozdíl od výše zmíněných, společnost Garmin nevyužívá žádný již mnoho let používaný programovací jazyk, ale vytvořila vlastní objektově orientovaný jazyk Monkey C [13]. Samotný jazyk bude popsán v kapitole 3. Operační systém zaštiťující vývoj na programovacím jazyku Monkey C je nazván Connect IQ [12]. Connect IQ stejně jako Android Wear a Movesense podporuje standardní technologie jako Bluetooth nebo Wi-Fi, na rozdíl od nich však má ještě možnost komunikovat skrze ANT+. Některé cyklopočítače Garmin komunikují skrze ANT+ s elektronickými sadami společnosti Shimano využívající technologii Di2 [37].

OS Connect IQ má další výhodu oproti konkurenci, a to ve výdrži baterie, která je uvedena na obrázku 1.4. Z tohoto obrázku je patrné, že zařízení, která využívají Android Wear, jsou v době nečinnosti vybíjeny až 10x rychleji, než je tomu u zařízení Garmin běžících na Connect IQ. Ze srovnávaných jsou na tom nejhůře zařízení společnosti Apple, jelikož v době nečinnosti vydrží pouze 18 hodin [71], oproti Connect IQ jsou tedy vybíjeny až 25x rychleji. Poslední zařízení porovnávané na obrázku 1.4 jsou zařízení využívající OS Pebble. U těchto zařízení výrobce udává výdrž baterie v době nečinnosti až 168 hodin (7 dní) [72]. Zařízení využívající Pebble jsou tak z porovnávaných OS nejbližší, stále je to však pouze jedna třetina

² Součástka starající se o výpočty hodnot z senzorů (GPS, ANT+ a další).

výdrže baterie oproti zařízením využívajícím OS Connect IQ. Některá zařízení Garmin dokáží v režimu nečinnosti vydržet až 14 týdnů jak je uvedeno v [73].



Obrázek 1.4: Srovnání ve výdrži baterie [22][50][51][71][72]

3 Monkey C

Aplikace na zařízeních Garmin používají objektově orientovaný programovací jazyk Monkey C popsaný v [13]. Tento jazyk vychází z programovacích jazyků C [28], Java [29], Javascript [30], Python [31], Lua [32], Ruby [33] a PHP [34]. Jednoduchý příklad vytvoření třídy *Kocka*, s vytvořením objektu *kitty* jménem Micka v Monkey C je uveden ve výpisu kódu 1.1.

```
class Kocka {  
    var jmeno;  
    function initialize(kocka) {  
        jmeno = kocka;  
    }  
}  
  
function main() {  
    var kitty = new Kocka("Micka");  
    System.print(kitty.jmeno); //vypíše do konzole Micka  
}
```

Výpis kódu 1.1. *Vytvoření třídy a objektu v Monkey C*

Stejně jako je tomu v programovacím jazyce Java, Monkey C kompiluje zdrojové soubory do byte kódu, který je následně interpretován virtuálním strojem Monkey Brains. Virtuální stroj Monkey Brains poté komunikuje s dalšími dostupnými API: pro grafiku, rozvržení displeje, lokaci, zaznamenávání aktivity, trvalé ukládání dat, komunikaci s mobilními zařízeními a senzory. Veškerá naměřená data je možné sdílet pomocí platformy Connect IQ skrze webové prostředí. Dále je možné přepisovat zdrojová data zařízení, nahrávat rozdílné obrázky, fonty a layout, nebo-li rozvržení displeje. Všechna tato zdrojová data se zvláště zapisují do XML konfiguračních souborů pro každý jednotlivý produkt. Pro účely testování existuje ve speciálním pluginu pro vývojové prostředí Eclipse simulátor, díky kterému jsme schopni testovat námi specifikovaný typ hodinek a je tak velmi jednoduché přepínat mezi dotykovými a tlačítkovými zařízeními. V době běhu programu je možné se dotazovat na systémová data aplikací. Monkey C aplikace jsou dynamicky linkovány a pokud aplikace odkazuje na API, které neexistuje na daném zařízení, tak aplikace spadne při běhu, chová se tedy jako interpretovaný jazyk. Monkey C dále používá podobně jako C++ `using statement`. Výrok `using` slouží k určení rozsahu jmenného prostoru.

Veškeré třídy a jejich metody lze najít v API dokumentaci viz [16]. Kořenovou třídou je `Toybox`, která obsahuje další podtřídy. Z názvů většiny tříd lze jednoduše odvodit, jaké metody sdružují a k čemu se používají. Narozdíl od jmenných prostorů v Javě, v Monkey C jsou třídy

statickými objekty, které mohou obsahovat metody a proměnné. Všechny tyto objekty jsou alokovány na haldě (heap) a o jejich uvolnění z paměti se stará virtuální stroj. Narozdíl od Javy, zde tak není použit Garbage Collector, ale objekty jsou uvolňovány pomocí reference counting. Rozdíly mezi těmito typy uvolňování paměti a jejich výhody jsou popsány v článku [38]. Monkey C nemá primitivní typy jako je tomu například u Javy, takže například integer a float jsou brány jako objekty. Navíc také překladač Monkey C neověřuje datové typy a místo toho nahlásí chybu při běhu programu, pokud daná metoda provádí operaci chybně. Příklad bezchybné metody je uveden ve výpisu kódu 1.2.

```
function add(a, b) {  
    return a + b;  
}  
  
function main() {  
    var a = add(1, 3); // vrátí 4  
    var b = add("Hello ", "World"); // vrátí "Hello World"  
}
```

Výpis kódu 1.2. *Příklad bezchybné metody v Monkey C.*

Jazyk Monkey C využívá programovací techniku založenou na objektech, takzvaný “Duck typing” koncept [39]. Samotné objekty jsou v Monkey C vázány na třídu, ve které byly vytvořeny, to znamená, že je mimo tuto třídu nemůžeme použít.

3.1 Překladač

Následující informace jsou čerpány z [14]. Zdrojový překladač kompiluje obrázky, text a statická data do adresáře zdrojů. V tomto adresáři jsou pro každý typ dat vytvořeny soubory ve formátu XML, které obsahují informace o těchto datech. Tyto informace následně využívá aplikace, která ke zdrojům přistupuje za běhu programu. V aplikaci je automaticky vygenerován modul (statický objekt, který může obsahovat metody, třídy a proměnné) pojmenovaný klíčovým slovem Rez. Tento modul obsahuje identifikátory, které se následně používají pro odkazování na zdrojové soubory. Na prostředky lze také odkazovat z jiného zdrojového souboru, a to použitím syntaxe @<module>.<id>, jejíž správné použití lze vidět ve výpisu kódu 1.3, kde parametr id nabývá hodnoty uzivatelsky a parametr module odpovídá hodnotě Strings (zdrojový soubor).

```
//vstupní soubor strings.xml
<string id="uzivatelsky">Menu item</string>

//použití
<menu id="MainMenu">

    <menu-item id="polozka1" label="@Strings.uzivatelsky" />

</menu>
```

Výpis kódu 1.3. *Příklad odkazování na zdroje.*

Zařízení Garmin mají rozdílnou velikost displeje a technologie zobrazování, tudíž bitmapy musí být konvertovány pro každé zařízení zvlášť. Zdrojový překladač nejprve vygeneruje zdroje pro všechny produkty. To nám umožňuje vytvořit skupiny produktů pro různé typy displejů: černobílé, barevné, kulaté a jiné. Jelikož každé zařízení disponuje rozdílným počtem barev, je nutné specifikovat paletu barev, která by se měla na obrázek použít. Definované barvy jsou poté namapovány v co nejpresnějším měřítku podle daného zařízení. Stejná bitmapa může v konečném důsledku vypadat na každém zařízení jinak.

Podporované formáty fontů jsou TXT a PNG. Jelikož bitmapové fonty mohou zabírat velké množství operační paměti, je defaultně nastavena konverze na 1-bitové fonty s vypnutým anti-aliasem nebo-li vyhlazováním. Pokud víme, že z daného fontu budeme používat pouze určité znaky, je možné tyto znaky vypsát pomocí parametru `filter` a tak ušetřit více paměti na zařízení.

3.2 **Datové úložiště**

Zařízení Garmin mají omezenou velikost vnitřního úložiště řádově v desítkách KB (Eclipse simulátor udává omezení velikosti paměti pro jednu aplikaci například u modelu fr235 na 64.4 KB), a tak by nebylo na nich možné provozovat aplikace, které potřebují uchovávat velké množství dat. Z tohoto důvodu vývojáři společnosti Garmin vytvořili tři možnosti, jak data uchovávat mimo vnitřní úložiště zařízení.

První z nich je úložiště reprezentující data zapsaná na fyzický disk připojeného zařízení (počítače nebo mobilního telefonu). Tato data jsou uchována i při selhání aplikace, ale jsou poskytovány pouze aplikaci samotné. Pro koncového uživatele to tedy znamená, že nemůže tato data dále upravovat. Uloženy mohou být pouze datové typy: number, float, long, double, char, string, boolean, array a dictionary. O ukládání dat se stará třída `Storage` formou páru klíč-hodnota, do dvojice souborů s koncovkami `.IDX` a `.DAT`. Pro nastavení hodnoty slouží metoda `setValue()`, pro její získání `getValue()`. Pár klíč-hodnota má maximální velikost 8 KB při celkové velikosti úložiště 128 KB.

Druhou možností uchování dat jsou konstantní hodnoty, nebo-li vlastnosti, které jsou definovány při sestavování aplikace a jsou zahrnuty do spustitelného souboru. Zde se ukládají zejména hodnoty, které by neměly být definovány přímo v kódu. Vlastnosti mohou dále určit

výchozí hodnoty nastavení aplikace. Pro uložení konstantních hodnot je nutné v adresáři zdrojů `settings` vytvořit soubor `properties` formátu XML definovat značku `<property>` s dvěma povinnými parametry `id` a `type` (`string`, `boolean`, `number` nebo `float`).

Třetí možností je uživatelsky editovatelné nastavení poskytované skrze aplikaci Garmin Connect [24] pro mobilní telefony nebo web a aplikaci Garmin Express [25] pro počítače. Tyto aplikace budou popsány níže v práci. Zejména pro typy aplikací uživatelské vzhledy hodinek a datová pole je toto jediný způsob, jak získat od koncového uživatele data na vstupu. Výchozí hodnoty v tomto nastavení určují vlastnosti. Nastavení slouží jako nástroj popisující jak dané vlastnosti zobrazit uživateli. Chceme-li toto nastavení definovat, je nutné v adresáři zdrojů `settings` vytvořit soubor `settings` ve formátu XML a použít párovou značku `<setting>`. Výpis kódu 1.4 ukazuje použití nastavení. Uživatel v nastavení aplikace uvidí jedno pole, které může vyplnit. Do tohoto pole musí (parametr `required` nastaven na hodnotu `true`) zapsat alfanumerický znak.

```
<settings>
    <setting propertyKey="@Properties.name" title="@Strings.name">
        <settingConfig type="alphaNumeric" required="true" />
    </setting>
</settings>
```

Výpis kódu 1.4. *Uživatel uvidí u aplikace možnost nastavení s jedním polem, které je navíc povinné vyplnit*

4 Platforma Connect IQ

Pomocí platformy Connect IQ můžeme pracovat se zařízením, které používá rozdílné senzory, funkce, typy aplikací, ANT+ [35] nebo dostupné API [52]. Mezi konfigurovatelné ANT+ senzory pro měření patří: tepová frekvence, teplota, kadence, rychlost, monitor napájení, řazení, foot pod (krokoměr) a mezi dostupná API patří: uživatelské rozhraní, kalendář, GPS, konektivita (komunikace se senzory) a vnitřní úložiště.

Pomocí Connect IQ SDK můžeme vytvářet několik typů aplikací jmenovitě: aplikace poskytující audio záznamy, datová pole, aplikace, vzhledy hodinek a doplňky nebo-li widgety. Všechny tyto typy budou detailně popsány v následujících odstavcích nebo se o nich můžete dočíst v [15].

Datová pole jsou speciální typ aplikace, který funguje v rámci podporované aktivity. Je to pole, které je možné přidat do rozložení polí dané aktivity (výchozí rozložení obrazovky má 4 pole). Je možné přizpůsobit vestavěné funkce a zobrazovat informace o aktivitě specifikovaným způsobem. Můžeme měnit výpočty dat a hodnot, které zařízení generuje a ukládá. Při návrhu datového pole je nutné klást důraz na zobrazení dané informace, je potřeba zvolit stejné fonty, velikost písma a grafiku, aby se zobrazované informace nepřekrývaly. Datová pole mohou pracovat se soubory .FIT (data dané aktivity) a mají přístup k ANT+ senzorům. Jedna z neznámějších a nejstahovanějších aplikací typu datová pole je dle Garmin Store IQ aplikace Strava Relative Effort [40], jenž je vidět na obrázku 1.5. Tato aplikace analyzuje srdeční tep během zapnuté aktivity a dle toho stanoví výsledné skóre (pole zobrazuje číselnou hodnotu).



Obrázek 1.5: *Strava Relative Effort* [40]

Typ **aplikace** je nejvíce používaný, umožňuje největší flexibilitu a má přístup ke všem senzorům, GPS, ANT+, které daný typ zařízení nabízí. Dále může manipulovat s .FIT soubory, které uchovávají data o uživatelských aktivitách. Aplikace má také možnost navázat spojení a komunikovat s mobilním telefonem pomocí Bluetooth. Hlavní nevýhodou aplikací je nejčastěji vysoká spotřeba baterie z toho důvodu, že většina aplikací využívá vestavěné senzory (GPS,

barometr a další) a na základě dat získaných ze senzorů zobrazují každou vteřinu nové informace. Při navrhování aplikace je nutné věnovat velkou pozornost potřebám uživatelů a případně jim poskytnout vhodnou zpětnou vazbu, metriky a nastavení skrze dostupné možnosti. Největší počet stažení má na Garmin Store IQ komplexní aplikace zobrazující mapovou navigaci s názvem dwMap [41], jejíž prémiovou verzi lze vidět na obrázku 1.6.



Obrázek 1.6: *Aplikace dwMap [41]*

Druh aplikace **vzhled hodinek** běží neustále a vyskytuje se na hlavní obrazovce hodinek. Nejčastěji slouží k zobrazení času, data a stavu baterie. Špatně navržená aplikace může mít velmi negativní vliv na výkon baterie daného zařízení. Hlavně z těchto důvodů jsou možnosti tohoto typu aplikace velmi omezené. Přístup má pouze ke grafice, bitmapám, fontům, stavu aktivit, stavu baterie a uživatelskému profilu. Aktualizace dat u aplikací vzhledu hodinek probíhá typicky jednou za minutu v režimu nízké spotřeby, ale může probíhat i každou sekundu. Dle Garmin Store IQ patří aplikace NoFrills [53] mezi nejstahovanější vzhled hodinek. Ukázku z této aplikace lze vidět na obrázku 1.7. NoFrills kromě času, data a stavu baterie dokáže zobrazovat také informační ikonky (zapnutý Bluetooth, počet zpráv na spárovaném mobilním zařízení a další).



Obrázek 1.7: *Aplikace NoFrills [53]*

Widget je typ aplikace, která je neinteraktivní a využívá data z hodinek nebo připojeného telefonu, k zobrazení například počasí, kalendáře, kompasu a další. Widget by měl rychle načíst data, ale také včas uvolňovat paměť, jinak je velice pravděpodobné, že na něj bude zařízení pomaleji reagovat a než zobrazí samotná data tak z obrazovky hodinek zmizí. Widget je viditelný v době používání a po určité době je pohled navrácen zpět na výchozí hlavní obrazovku. Stejně jako aplikace má přístup ke všem senzorům, ANT+ i GPS a je schopen navázat spojení s mobilním telefonem pomocí Bluetooth. Mezi základní widgety patří Sun & Moon [68] zobrazující data související s časy východu/západu slunce a měsíce. Ukázku lze vidět na obrázku 1.8.



Obrázek 1.8: *Sun & Moon widget [68]*

Aplikace poskytující audio záznamy je prakticky velmi podobný typ jako aplikace. Navíc je však možné synchronizovat audio záznamy z aplikací třetích stran, spravovat stažený audio obsah a přizpůsobit přehrávání zvuku ve svém zařízení. Jedná se o nejmladší typ aplikace a byl zaveden s příchodem zařízení jako Forerunner 645 Music [43] nebo vivoActive 3 Music [42]. Známou aplikací, která se dostala i na Garmin Store IQ je Spotify [44], která slouží k přehrávání hudby.

Aplikace mohou být vytvořeny také jako služba běžící na pozadí. Služby jsou využívány k zachycení události například splnění uživatelského cíle, splnění počtu kroků a další. Pokud je spuštěna služba na pozadí, je volána metoda `getServiceDelegate()`, která vrací systémového delegáta a vyvolá metodu odpovídající spuštěné události. Jakmile služba spuštěná na pozadí dokončí provádění činnosti, zavolá se metoda `exit()` a služba se ukončí. Služby mohou být ukončeny za účelem uvolnění paměti nezbytné pro chod aplikace nebo pokud nejsou správně ukončeny v rámci 30 sekund od otevření. Každá aplikace také musí obsahovat třídu `Application`, která slouží jako vstupní bod dané aplikaci.

Pro správný chod všech výše zmíněných typů aplikací je zapotřebí konfigurační soubor, který obsahuje všechny potřebná data o aplikaci.

4.1 Konfigurační soubor

Všechny Connect IQ aplikace vyžadují zdrojový konfigurační soubor nazvaný manifest, viz také [17], který je ve formátu XML. Soubor obsahuje konfigurační informace potřebné k sestavení aplikace. Jmenovitě zde nalezneme typ aplikace, podporované zařízení, jazykové mutace a další. Soubor manifest je generovaný automaticky vývojovým prostředím. Ve výpisu kódu 1.5 je ukázán vzorový soubor manifest.xml, který je popsán dále.

```
<iq:manifest xmlns:iq="http://www.garmin.com/xml/connectiq"
version="3">

  <iq:application entry="TestApp" id="230d37bc-72e4-447a-a146-
b3012f7de044" launcherIcon="@Drawables.LauncherIcon"
name="@Strings.AppName" type="widget" version="1.0.0">

    <iq:products>
      <iq:product id="fr235"/>
      <iq:product id="vivoactive_hr"/>
    </iq:products>
    <iq:permissions>
      <iq:uses-permission id="Sensor"/>
    </iq:permissions>
    <iq:languages>
      <iq:language>ces</iq:language>
      <iq:language>eng</iq:language>
    </iq:languages>
  </iq:application>
</iq:manifest>
```

Výpis kódu 1.5. *Vzorový soubor manifest.xml se základním nastavením*

Element `application` obsahuje následující informace. Atribut `id`, což je jedinečný 128bitový UUID identifikátor³ generovaný používaným IDE. Atribut `entry`, který specifikuje objekt naší aplikace, dále atribut `name` pro jméno aplikace a `launcherIcon` pro ikonu aplikace (pokud není specifikována, je použita defaultní ikona). Atribut `type` určuje jaký typ aplikace se bude kompilovat. V nejnovější verzi Connect IQ SDK je pět typů: `watchface` (vzhled hodinek), `datafield` (datová pole), `widget`, `watch-app` (aplikace) a `audio-`

³ Případně je možné vygenerovat jej ručně na <https://www.uuidgenerator.net/version4>.

`content-provider-app` (aplikace poskytující audio záznam), které odpovídají typům aplikací. Dle zvoleného typu aplikace v souboru manifest určíme, kde bude aplikace umístěná na daném zařízení a jaké API bude moci využívat. Tag `<product>` určuje jaká zařízení budou podporována. Existují dva imaginární produkty určené pro obecné prototypování `round-watch` a `square-watch`. První zmíněný zahrnuje skupinu zařízení obsahující pouze tlačítka, zatímco druhý dotykové obrazovky se základními tlačítky. Pokud bude naše aplikace používat moduly jako zaznamenávání aktivity, komunikace, polohování, senzory nebo uživatelské profily, je nutné získat oprávnění pro používání těchto modulů od uživatele při samotné instalaci aplikace. Tento seznam požadavků se následně zapisuje do souboru manifest tagem `<uses-permission>`.

4.2 Kvalifikátory

Následující kapitola čerpá informace z [18]. Zdrojové soubory lze psát dle konkrétního typu nebo skupiny zařízení, podle validní jazykové mutace nebo také dle určitého názvu zařízení. Pro toto nastavení použijeme takzvané kvalifikátory, což jsou klíčové slova udávající právě jednu ze zmíněných kategorií. Skupinovými kvalifikátory rozumíme dvě skupiny, ty dělíme podle tvaru displeje (čtvercový, kulatý, obdélníkový) a velikosti displeje v pixelech (148x205, 218x218 a další). Kvalifikátory se také dají mezi sebou řetězit za pomoci pomlčky, která se vkládá mezi tato dvě klíčové slova. Kvalifikátory se dále mohou přidávat do základní zdrojové složky, ze které pak podsložky automaticky dědí specifikované kvalifikátory. Pokud kompilujeme kód na zařízení, pro které jsme zdroje předem neurčili, pak toto zařízení bude využívat defaultní zdrojovou složku. Kvalifikátory je možné kombinovat mezi sebou, nicméně pro kombinování existují určité pravidla. Kvalifikátory skupiny a specifického zařízení zároveň nelze kombinovat, tudíž například `resources-square-fr235` nebude fungovat a kompilátor zadanou složku ignoruje, protože je porušeno právě toto pravidlo. Dalším pravidlem, na které musíme dát pozor, je posloupnost mezi kvalifikátory. Pokud je použit kvalifikátor konkrétního zařízení, pak je nadřazený těm skupinovým. Pokud chceme definovat zdrojové soubory pro zařízení na základě velikosti displeje, je vždy nutné uvést tvar displeje (povinný parametr), tedy například `resources-square-148x205`. Priorita použití zdrojových souborů se poté bere od více specifických definic k méně specifickým definicím. V případě, že jsou zdrojové složky vymezovány podle jazykové mutace, řídíme se normou ISO 639-2 [1]. Jazykové kvalifikátory můžeme kombinovat s oběma zmíněnými typy kvalifikátorů a uvádíme je v názvu vždy jako poslední. Pro zařízení Forerunner 235 s českou jazykovou mutací by byla složka pojmenována `resources-fr235-ces`.

Kvalifikátory se dále používají v takzvaných jungle (konfigurační soubor, obsahující instrukce, kvalifikátory a proměnné) souborech, které jsou automaticky vygenerovány vývojovým prostředím při založení nového projektu. Je nutné, aby tento soubor obsahoval každý projekt. Pokud bychom soubor smazali, nastane chyba v programu. Pomocí tohoto souboru, který se musí nacházet v kořenovém adresáři projektu a má příponu `.jungle`, je možno vytvářet vlastní instrukce pro sestavení aplikací. Lze také definovat cesty k zdrojovým souborům dle dané

skupiny nebo typu zařízení a specifikovat uživatelské knihovny tkzv. Monkey Barrels. Po sestavení projektu se jako první aplikují instrukce z výchozího a poté uživatelského jungle souboru, pokud je v projektu vytvořen. Instrukce použité ve výpisu kódu 1.6 demonstrují rozdělení zdrojových souborů dle tvaru displeje. Překladač tedy při sestavování aplikace na zařízení s kulatým displejem hledá klíčové slovo `resource-round` a ostatní ignoruje.

```
# konfigurování cesty k roztržiděným zdrojovým souborům
round.resourcePath = $(base.resourcePath);resource-round
semiround.resourcePath = $(base.resourcePath);resource-semiround
rectangle.resourcePath = $(base.resourcePath);resource-rectangle
```

Výpis kódu 1.6. *Ukázka využití kvalifikátorů pro zdrojové soubory*

Instrukce ve všech jungle souborech, tedy i ty ve výchozím souboru, jsou dále editovatelné a mohou obsahovat kvalifikátory, lokální proměnné, hodnoty a komentáře. Správně napsané instrukce jsou tvořeny kombinací kvalifikátor-hodnota nebo proměnná-hodnota.

4.3 API pro práci s uživatelským rozhraním

Tato kapitola čerpá ze zdroje [20]. Garmin zařízení jsou velmi variabilní a navrhnout jedno komplexní uživatelské rozhraní pro různá zařízení je poměrně náročné. Některá zařízení mají fyzická tlačítka, jiná jsou ovládána pouze pomocí dotyku. Connect IQ SDK nám umožňuje přenášet uživatelská rozhraní mezi rozdílnými produkty, pomocí rozložení definovaném v souboru formátu XML. Na zdrojové úrovni máme tedy možnost definovat jak rozložení uživatelského rozhraní, tak dále specifikovat pro jaké zařízení se má použít.

Uživatelské rozhraní reprezentuje třída `View`, která funguje na principu LIFO⁴ zásobníku a obsahuje metody v závislosti na typu aplikace. Pro widgety a aplikace existují ve třídě `View` následující metody: `onStart()` je volána při zapnutí aplikace, `onLayout()` pro načtení zdrojů uživatelského rozhraní, `onShow()` je metoda volána v případě, že se aplikace dostane do popředí, `onUpdate()` pro aktualizování třídy a `onHide()`, pokud je aplikace odebrána z obrazovky. Pro vzhledy uživatelské obrazovky a datová pole jsou některé metody nepoužitelné, zde třída `View` tedy obsahuje pouze metody `onLayout()`, `onShow()` a `onUpdate()`.

Vstupním bodem u všech typů aplikací je metoda `onLayout()`, která je zavolána jako první a načte potřebné zdroje (bitmapy, texty a další). Pokud se dostává aplikace do popředí, je volána metoda `onShow()`. Následně v případě, že je aplikace aktivní a nachází se v popředí, je volána metoda `onUpdate()`, která slouží k dynamickým změnám obsahu. Pro vzhledy

⁴ Z angličtiny Last in First Out (poslední dovnitř první ven), metoda řízení dat.

uživatelské obrazovky je tato metoda volána každou minutu, pro datová pole každou sekundu. Pokud třída neimplementuje metodu `onLayout()`, potom je jakýkoliv zdroj vykreslen automaticky.

4.4 Vykreslovací kontext

Tato kapitola čerpá ze zdroje [20]. K vykreslování grafiky se používá třída `DC`, která obsahuje několik metod pro vykreslování, včetně metod pro vykreslení základních geometrických útvarů jako například `drawCircle()` pro kruh, `drawPoint()` pro bod, `drawText()` pro text a další. Tato třída obsahuje také metodu `clear()` pro vyčištění obrazovky, která nastaví všechny pixely nevyskytující se ve vykreslovaných tvarech na barvu pozadí a tím dojde k jejich zneviditelnění. Pro nastavení barvy vykreslitelné grafiky slouží metoda `setColor()`. Barvy jsou do metody předávány ve formě `0xRRGGBB`, kde dvojice písmen znázorňuje aditivní míchání barev⁵ RR-červená, GG-zelená a BB-modrá. Při nastavování těchto barev se zařízení snaží vybírat z nejbližších dostupných barev v systému, jelikož jak bylo popsáno výše, každé zařízení disponuje rozdílným počtem podporovaných barev a na každém zařízení tedy může mít výsledný objekt odlišnou barvu. Pokud chceme z nějakého důvodu vykreslovat pouze do určitého prostoru, API uživatelského rozhraní nám nabízí speciální metodu, pomocí které toho můžeme docílit. Metoda se jmenuje `setClip()` a přijímá 4 parametry jmenovitě šířku, výšku v pixelech a souřadnice `x, y` udávající kde se má plocha nacházet. Pro vykreslování uživatelských fontů a textů se používá metoda `drawText()`. Pro její správné použití je třeba specifikovat 5 parametrů souřadnice `x,y`, font (může být uživatelský nebo systémový), text a zarovnání (`Graphics.TEXT_JUSTIFICATION_XXX`, kde `XXX` může nabývat hodnot `left`, `right` a `center`).

4.5 Manipulace se vstupy

Tato kapitola čerpá ze zdroje [20]. Uživatelský vstup umožňuje uživatelům manipulovat a reagovat na výstup z aplikací. V zařízeních Garmin se uživatelské vstupy dělí do dvou kategorií. První kategorie obsahuje zařízení využívající dotykový displej a druhá kategorie zařízení, které disponují fyzickými tlačítky.

Manipulací s uživatelským vstupem se zabývá třída `InputDelegate`. Delegát je objekt této třídy, který implementuje určité metody specifické pro manipulaci se vstupem. API uživatelského rozhraní nabízí metody jak pro dotyková zařízení, tak pro tlačítková. Při psaní uživatelské aplikace pro tlačítková zařízení je tedy velmi jednoduché rozšířit chování aplikace tak, aby prováděla danou činnost i na dotykových zařízeních. Z pozice vývojáře je nutné se pokaždé zamýšlet, na jaký druh aktivity bude daná aplikace používána. Existuje totiž skupina aktivit, při kterých nelze použít dotykový displej například plavání, kanoistika a další.

⁵ Způsob míchání barev, kdy se jednotlivé složky barev sčítají a vytváří světlo větší intenzity.

Třída spojená s obsluhou uživatelského vstupu se jmenuje `BehaviorDelegate`. Tato třída rozšiřuje již zmíněnou třídu `InputDelegate` a stará se konkrétně o obsluhu událostí. `BehaviorDelegate` mapuje vstupy na běžné operace skrze všechna zařízení, tudíž zařadí stejnou funkcionalitu fyzického tlačítka pro přjetí doleva, jako když u dotykového zařízení uživatel přetáhne prstem doleva. Tyto interakce jsou ve třídě rozděleny do jednotlivých metod s návratovou hodnotou `boolean` a vrací hodnotu `true` nebo-li pravda v případě, že dojde k jejich zachycení. Obsahuje tedy například metody `onNextPage()`, `onBack()`, `onMenu()` a další.

Obecně by se však komplikované vstupy uživatelů a zpětná vazba neměly provádět přímo na daném zařízení, ale raději prostřednictvím webové aplikace nebo mobilního telefonu (zařízení pak nemusí uchovávat data spojená s obsluhou těchto vstupů). Jsou zde tři možnosti jak se vypořádat se vstupy a dvě možnosti, jak může být koncovému uživateli poskytnuta zpětná vazba. Vstupy se dají řešit pomocí nabídky menu, generického výběru nebo číselného výběru (číselný výběr se v nejnovější verzi SDK již nepoužívá a je označený jako zastaralý, náhradou je zmíněný generický výběr). Zpětná vazba se pak dá řešit dialogy, konkrétně potvrzovacím dialogem a dialogem průběhu.

4.5.1 Nabídka Menu

Nabídka menu je seznam možností, které jsou uživateli nabízeny. Menu je zobrazeno v závislosti na typu použitého zařízení (příklad zobrazení menu na zařízení `fr235` lze vidět na obrázku 1.9), jednotlivé položky jsou však vždy definovány ve zdrojovém souboru ve formátu XML, jak lze vidět ve výpis kódu 1.7. Následně lze na tyto položky odkazovat, viz výpis kódu 1.8.

```
<menu id="MainMenu">
    <menu-item id="polozka1" label="@Strings.menu_item_1_label" />
    <menu-item id="polozka2" label="@Strings.menu_item_2_label" />
</menu>
```

Výpis kódu 1.7. *Příklad menu s dvěma položkami.*



Obrázek 1.9: *Ukázka menu na zařízení `fr235`*

Z poskytnutého XML souboru je následně vygenerován objekt tohoto menu. Pro jeho použití je nutné v odpovídající třídě poslat objekt menu a jeho delegáta do popředí. K tomu slouží metoda `pushView()`, která akceptuje tři parametry, kde první parametr je naše vytvořené menu, druhý parametr zmíněný delegát a třetí parametr přechod nebo-li způsob zobrazení (například `SLIDE_XXX`, kde `XXX` reprezentuje hodnoty `left`, `right`, `down` a další).

```
class MenuDelegate extends Ui.MenuInputDelegate {  
    function onMenuItem(item) {  
        if ( item == :polozka1 ) {  
            // proved  
        } else if ( item == :polozka2 ) {  
            // proved  
        }  
    }  
}
```

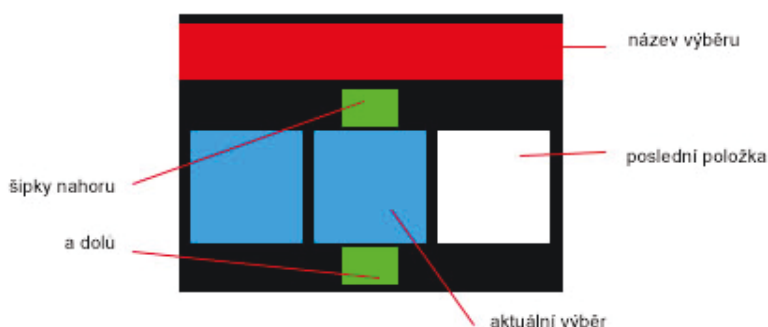
Výpis kódu 1.8. *Použití menu v aplikaci*

Pro komplexnější nabídky menu Connect IQ ve verze 3.0 nabízí třídu `Menu2`. Tato třída obsahuje nové metody, které umožňují vytvořit dynamicky měnící se položky, grafické texty, zaškrtačací pole nebo-li checkboxy a další elementy menu.

4.5.2 Generický a číselný výběr

Pro implementaci generického výběru je nutné použít třídy `Picker`, `PickerDelegate` a `PickerFactory`. Tyto třídy umožňují na obrazovkách zařízení vytvářet seznamy, mezi kterými si uživatel může vybírat. Generický výběr obsahuje samotné objekty, nadpisy, šipky vpřed, vzad a potvrzovací tlačítko. Přičemž vzhled šipek vpřed, vzad a potvrzovacího tlačítka může být dále modifikován. Na obrázku 1.10 je příklad generického výběru pro hranatý displej, kde červený horní panel reprezentuje název výběru, zelené pole reprezentuje šipky nahoru a dolů (pro pohyb mezi nabídkami) a čtverce znázorňují výběry, přičemž bílý čtverec označuje poslední položku.

Číselný výběr je metoda, která by se již neměla používat, nicméně ve starších verzích Connect IQ SDK než je verze 3.0, umožňoval tento výběr upravovat nebo přiřazovat číselné datové typy. Nyní je nahrazen generickým výběrem, který je více univerzální a kromě číselných výběrů může vytvářet i výběry dle barvy, textu a další.



Obrázek 1.10: Ukázka generického výběru pro hranatý displej

4.5.3 Potvrzovací dialog a dialog průběhu

Potvrzovací dialog poskytuje jednoduchou obrazovku s možnostmi ano/ne. Tento dialog je ve své podstatě nejčastěji využíván při ukončení aktivity, kdy nám dialog na zařízení nabídne dotaz, zda má ukončenou aktivitu uložit či nikoliv.

Dialog průběhu se dá chápat jako obrazovka, která má dva stavy. První stav ukazuje dokončení nějaké akce a druhý stav pouze reprezentuje čekací obrazovku s vizualizovaným načítacím polem 0-100% (typicky při instalování aktualizací do hodinek). Vzhled a fungování načítacího pole je závislý na typu použitého zařízení.

4.6 Pozicování a senzory

Informace o pozicování a senzorech jsou čerpány ze zdroje [21]. Určování polohy stejně jako podporované senzory (GPS, barometr, měření tepové frekvence a další), patří k nejvyhledávanějším funkcím chytrých hodinek.

Souřadnice jsou uloženy ve formě klasického souřadnicového systému EU [70] nebo MGRS (z angl. Military Grid Reference System) používaný Severoatlantickou aliancí [69]. Zařízení Garmin získávají informace o poloze jak z družic americké GPS, tak ruských družic GLONASS.

Veškeré údaje ze senzorů je možné zaznamenávat do souborů s příponou .FIT, což je velmi užitečné při samotném vývoji a následné simulaci činnosti. Před jejich získáním je však v aplikaci nutné dané senzory povolit (nastavení v konfiguračním souboru manifest) a dále ve třídě `ActivityRecording` vytvořit objekt `Session`, pomocí metody `createSession()`. Poté již stačí zavolat metodu `start()`, která začne zaznamenávat data z povoleného senzoru. Pokud nastane úsek, který má být přeskočen nebo jsou nahrána všechna data, volá se metoda `stop()`. Zaznamenávané informace nejsou ukládány automaticky, nýbrž je nutné zavolat metodu `save()`, případně `discard()` pro zahození dat.

5 Specifikace zadání

Cílem bakalářské práce je nastudování platformy Garmin Connect IQ, navrhnutí jednotlivých komponent uživatelského rozhraní zobrazující data ze senzorů zařízení (stav baterie, srdeční tep a další) a zejména vytvoření konfigurovatelné aplikace hodinek Garmin.

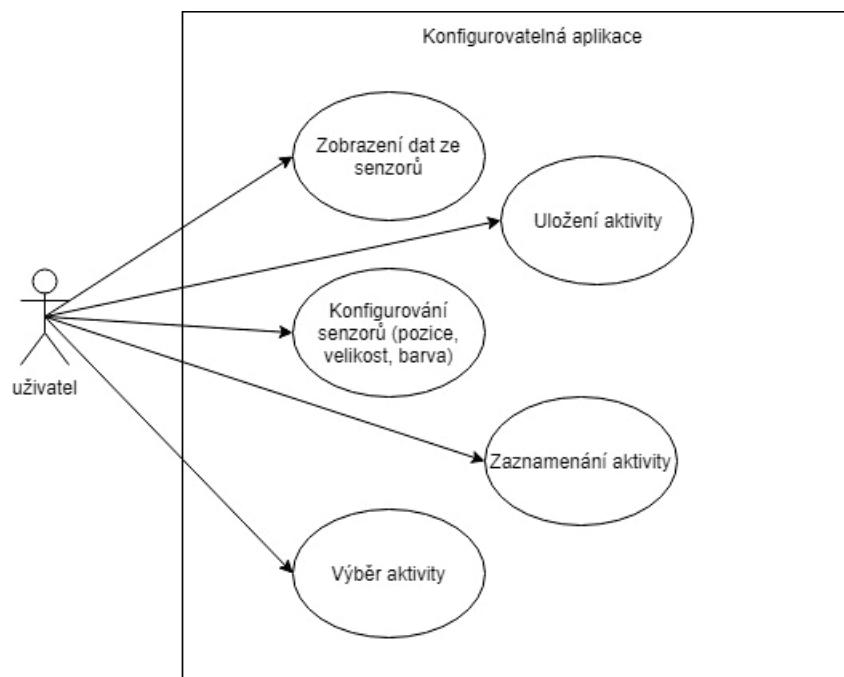
Z této specifikace vyplývá, že pro splnění zadání bude nutné navrhnout a vytvořit aplikaci, která bude moci zobrazovat konfigurovaná data uživatelem. Konfigurací dat chápeme jako možnost definování pozic, velikosti a barvy zobrazovaných dat, nebo-li metrik. Strukturu konfigurovatelné aplikace jsem rozdělil dle typů aplikací do dvou skupin. V první skupině popíši samotný typ aplikace, která se liší svou nadstavbou zvolení profilů aktivit (běh, kolo nebo turistika) a v druhé skupině popíši typy vzhledu hodinek a widget, z toho důvodu, že zde probíhal návrh a implementace totožně.

Samotný typ aplikace bude obsahovat úvodní obrazovku s názvem aplikace a pokynem k výběru aktivity. Diagram případů užití je zobrazen na obrázku 1.11. Po stisknutí tlačítka start se uživatel dostane na možnost výběru ze tří profilů aktivit běh, kolo nebo turistika a až po výběru dané aktivity se dostane k samotnému zobrazení konfigurovaných metrik, které bude možné dále zaznamenávat a uložit do zařízení.

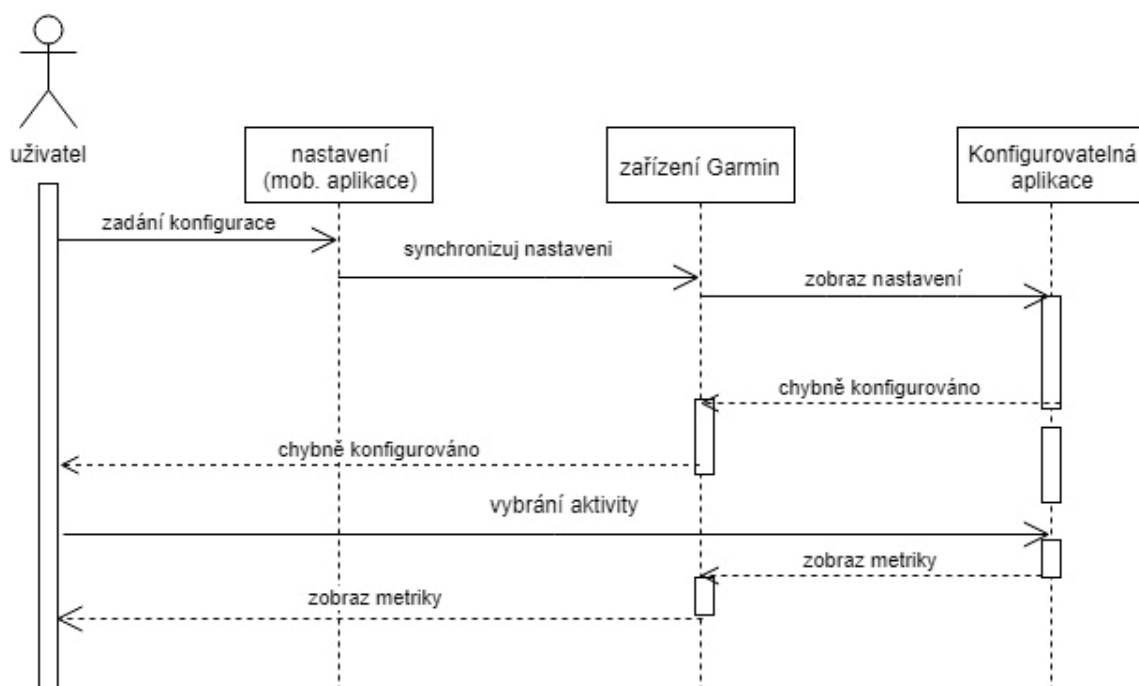
Konfigurovatelná aplikace typu vzhledu hodinek a widgetu bude ihned po spuštění zobrazovat výsledné rozložení metrik, bez možnosti zaznamenávání a uložení, jelikož to v těchto typech aplikací není možné. Úvodní obrazovka je u těchto typů aplikací nežádoucí z toho důvodu, že na ní uživatel nemá možnost reagovat.

Uživatel bude mít u všech typů aplikací možnost konfigurovat zobrazované metriky. Pro konfigurování se uživatel musí připojit k mobilnímu telefonu s nainstalovanou aplikací Garmin Connect Mobile [24] nebo k počítači s nainstalovanou aplikací Garmin Express [25], kde má možnost nastavení aplikace. Uživatel si může zvolit celkový počet zobrazovaných metrik (srdeční tep, počet kroků, rychlost a další) nebo-li velikost matice, barevné provedení, pozici daných metrik a jejich výslednou velikost na obrazovce.

Sekvenční diagram na obrázku 1.12 znázorňuje vzájemnou interakci uživatele s objekty nastavení v mobilní aplikaci, zařízení Garmin a konfigurovatelnou aplikací. Vzájemnou interakci zahajuje uživatel zadáním požadované konfigurace, týkající se zobrazení metrik do nastavení mobilní aplikace. Z mobilní aplikace jsou hodnoty následně poslány zprávou synchronizuj do zařízení Garmin. Po synchronizaci si konfigurovatelná aplikace uloží jak dané metriky zobrazit. V případě špatné konfigurace je uživatel informován zprávou chybně konfigurováno a musí postup provést znova. V opačném případě si může zvolit aktivitu (běh, kolo nebo turistiku) a poté jsou mu zobrazeny konfigurované metriky.



Obrázek 1.11: Diagram případu užití konfigurovatelné aplikace, použitý nástroj draw.io [54]



Obrázek 1.12: Sekvenční diagram znázorňující interakci uživatele s aplikací, použitý nástroj draw.io [54]

6 Analýza a návrh

Obsahem této části práce bude analýza a funkční požadavky konfigurovatelné aplikace. Pro správnou funkčnost aplikace je nutné definovat výslednou podobu a životní cyklus aplikace, která následně poslouží k implementaci konfigurovatelné aplikace dle stanovených požadavků práce. Prezentované poznatky budou vycházet z teorie popsané výše v práci.

6.1 Funkční požadavky

- Uživatel musí mít možnost konfigurovat velikost rozložení zobrazovaných metrik. V nastavení aplikace musí být editovatelné pole, do kterého bude uživatel zapisovat velikost rozložení (2x2, 3x3, 1x3 a další).
- Aplikace by měla podporovat zobrazení více druhů metrik najednou. V nastavení aplikace by měla být možnost výběru a zobrazení více metrik.
- Uživatel musí mít možnost měnit rozměry zobrazované metriky. Aplikace uživateli musí umožnit určit velikost zobrazované metriky v nastavení.
- Uživatel musí mít možnost výběru pozice zobrazované metriky. Aplikace uživateli musí umožnit stanovit výslednou pozici zobrazované metriky na obrazovce.
- Uživatel by měl mít při spuštění aplikace možnost výběru z více druhů aktivit.
- Aplikace umožní uživateli zaznamenávat údaje z metrik do zařízení.
- Aplikace by neměla metriky překreslovat přes sebe, ale vypsát v tomto případě chybu na obrazovku.
- Zobrazované metriky v aplikaci musí být čitelné.
- Aplikace by měla být vícejazyčná.

6.2 Seznam podporovaných metrik

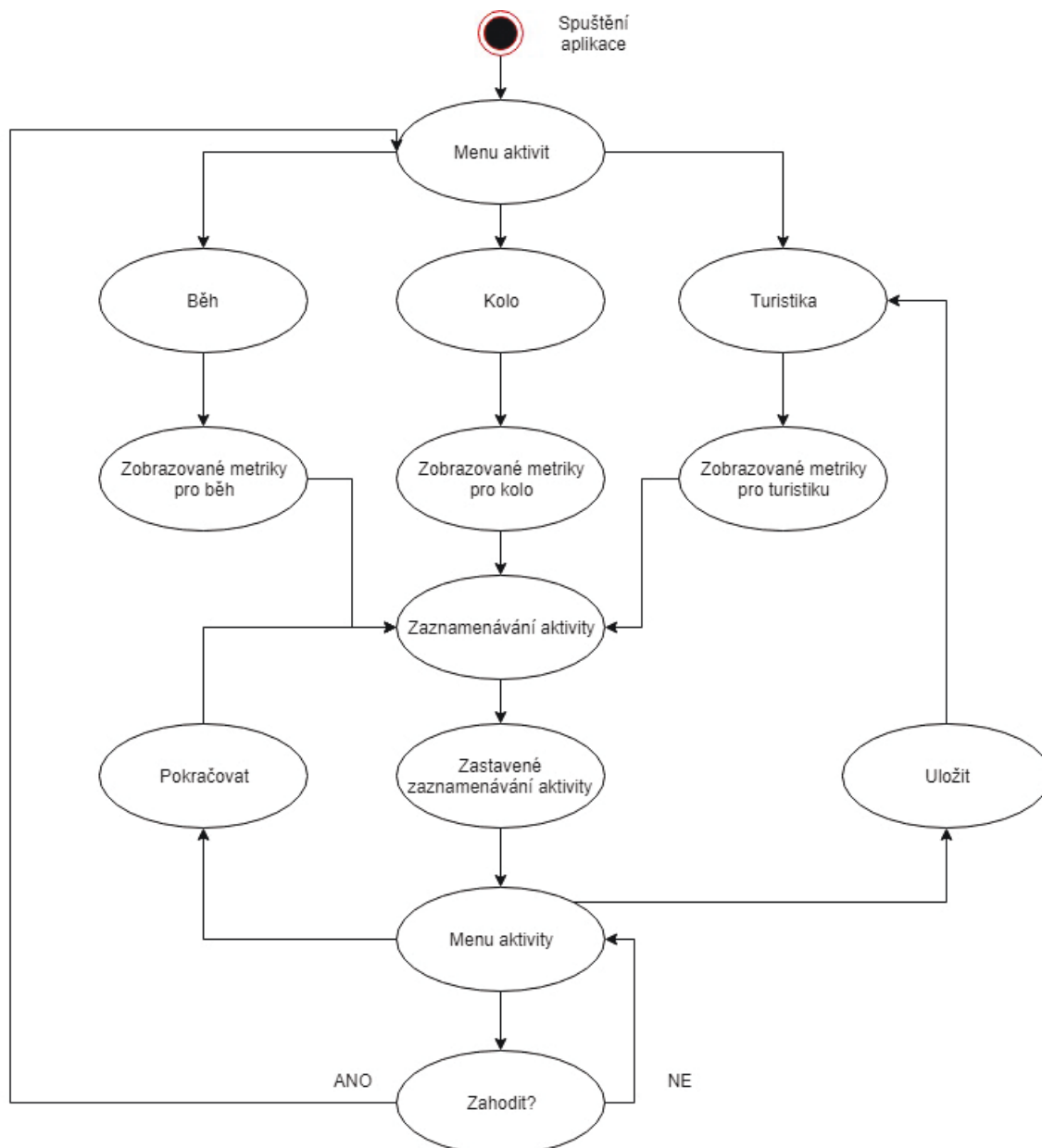
V aplikaci bude možné v závislosti na typu zařízení konfigurovat a zobrazit tyto metriky (v závorce udávám v jakých jednotkách budou zobrazeny):

- Aktuální čas
- Datum
- Počet kroků
- Cíl splnění počtu kroků (%)
- Srdeční tep
- Stav baterie (%)
- Spálené kalorie (kcal)
- Celková vzdálenost (km)
- Aktuální rychlost (km/h)
- Průměrná rychlost (km/h)
- Zdolaná vzdálenost během aktivity (km)
- Uplynulý čas aktivity

- Nastoupáno pater
- Sestoupáno pater
- Nastoupána nadmořská výška (m)
- Sestoupána nadmořská výška (m)
- Celková paměť zařízení (KB)
- Využitá paměť zařízení (KB)
- Volná paměť zařízení (KB)

6.3 Stavová analýza aplikace

Ukázka životního cyklu konfigurovatelné aplikace typu aplikace, lze vidět na obrázku 1.13. Po spuštění aplikace a posunutí z úvodní obrazovky, se uživatel dostane do nabídky menu aktivit, kde má možnost výběru profilu aktivity, kterou chce zaznamenávat. Z nabízených možností má na výběr aktivity: běh, kolo nebo turistiku. Po vybrání konkrétního profilu je uživateli zobrazena obrazovka s konfigurovanými metrikami tohoto profilu, které si uživatel nastavil v aplikaci Garmin Connect Mobile nebo Garmin Express. Následně dalším potvrzením, respektive stisknutím tlačítka start, se zapne nahrávání aktivity. Při stisknutí tlačítka zpět se aplikace chová přesně tak, jako defaultní Garmin aplikace (například pro běh) a zaznamená kolo, nebo-li úsek aktivity. Pomocí této funkce se dá jedna zaznamenávaná aktivita členit na různé dlouhé úseky v závislosti na tom, kolikrát je tlačítko zpět stisknuto. Pokud uživatel během aktivity opětovně stiskne tlačítko start, aktivita se pozastaví a uživateli je zobrazeno menu s možnostmi pokračovat, uložit nebo zahodit zaznamenanou aktivitu. V případě, že si uživatel vybere možnost pokračovat, tak se aktivita opět spustí v úseku, kde skončila a nadále zaznamenává údaje z metrik. Pokud uživatel vybere možnost uložit, tak se data o zaznamenávané aktivitě uloží do zařízení a pohled je navrácen na obrazovku s původně vybraným druhem aktivity. V posledním případě, při zvolení položky zahodit je uživateli zobrazen potvrzovací dialog s možnostmi ano/ne, popsány v kapitole 4.5.3. Při potvrzení dojde k zahození aktivity nebo-li smazání dat o zaznamenávané aktivitě a pohled je navrácen do menu s výběrem aktivit. Při zamítnutí je uživatel vrácen zpět na původní obrazovku menu s možnostmi, co s danou aktivitou udělat. V každém kroku této aplikace, kromě aktivního zaznamenávání aktivity, slouží tlačítko zpět k vrácení na předchozí obrazovku, a tedy i k vypnutí celé aplikace. Aplikace typu vzhled hodinek a widget slouží pouze k zobrazení konfigurovaných metrik, tudíž se dají v tomto diagramu chápat jako jeden stav zobrazované metriky pro běh, kolo nebo turistiku.

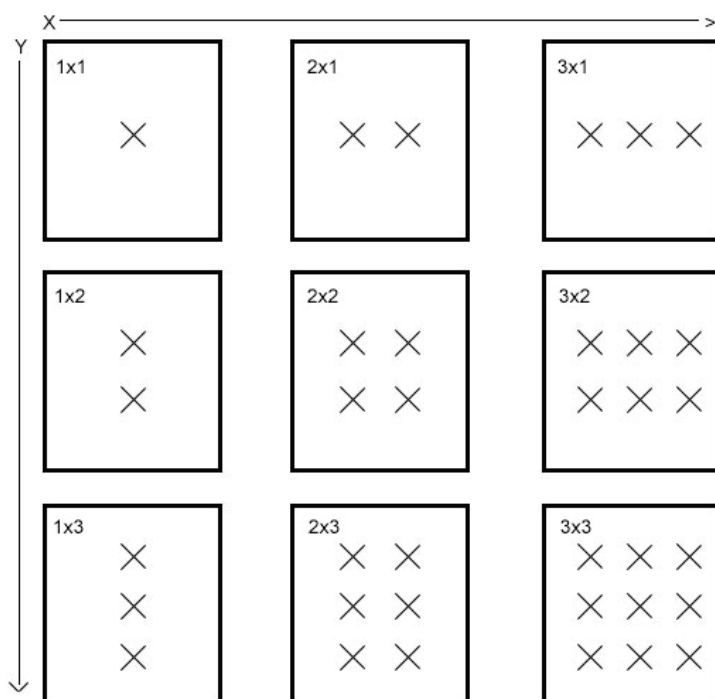


Obrázek 1.13: Životní cyklus konfigurovatelné aplikace, použitý nástroj draw.io [54]

6.4 Návrh uživatelského rozhraní

Veškeré nastavení bude uživatel provádět před spuštěním aplikace v Garmin Connect Mobile nebo Garmin Express. Konfigurovatelná aplikace bude podporovat rozložení o maximální velikosti 3x3 prvky, tedy 9 pozic pro metriky (srdeční tep, počet kroků, vzdálenost a další), seřazené od levého horního rohu do pravého spodního rohu zobrazované matice. Na obrázku 1.14 lze vidět všechny velikosti výsledné matice. Toto rozložení je dáno souřadnicemi X a Y, respektive XxY (při nastavení velikosti 3x2 bude rozložení vypadat jako obdélník s 6 křížky,

znázorněný na obrázku 1.14 ve třetím sloupci a druhém řádku), kde X znázorňuje počet metrik na vodorovné ose a Y znázorňuje počet metrik na svislé ose. Uživatel bude mít možnost na vstupu definovat velikost, pozici a barvu zobrazované metriky. Barvu bude moci uživatel volit ze čtrnácti základních barev zařízení. Ve výsledném zobrazení bude u některých metrik dostupná také grafická ikonka (srdce u srdečního tepu, plamen u počtu spálených kalorií a další). Uživatel bude definovat velikost zobrazení výsledné metriky v nastavení pomocí zadaného intervalu dvou čísel oddělených pomlčkou. První zadané číslo bude reprezentovat levý horní roh a druhé číslo pravý dolní roh zobrazované metriky. V závislosti na uživatelem zadaném intervalu pak bude zobrazovaná metrika roztažená přes několik pozic, orientovaná na výšku nebo na šířku, případně bude výsledná plocha v matici zabírat čtvercové rozložení. Pozice konfigurované metriky bude vždy uprostřed této plochy v případě zadání intervalu, nebo přímo na zadané pozici v případě zadání jednoho čísla uživatelem. U typu aplikace bude nastavení pozic a velikosti metrik rozdělené do tří kategorií přesně dle zvolených aktivit (běh, kolo nebo turistika) a bude tedy možné definovat různé rozložení a následné zobrazení metrik pro každou zvolenou aktivitu. Uživatel bude mít možnost nastavit zobrazení času na pozici 1 u aktivity běh, a u aktivity kolo například na pozici 7-9 a tak dále. V aplikaci budou dvě nabídky menu a jeden potvrzovací dialog. První nabídka menu bude sloužit k výběru aktivity a druhá nabídka menu pro zpracování zaznamenané aktivity. Obě tyto nabídky menu budou vypadat jako na obrázku 1.9. Potvrzovací dialog bude uživateli zobrazen v případě, že bude požadovat zaznamenanou aktivitu zahodit. Pokud uživatel zaznamená aktivitu a následně ji uloží do svého zařízení, bude mít možnost zobrazit údaje z těchto zaznamenaných metrik po synchronizování v mobilní aplikaci Garmin Connect Mobile, zmíněné výše.



Obrázek 1.14: Rozložení pozic pro metriky

7 Implementace konfigurovatelné aplikace

Tato kapitola se zabývá použitými nástroji k vytvoření konfigurovatelné aplikace, definování zjištěných problémů a detailním popisem implementace. Konfigurovatelná aplikace, již jsem vytvořil, je stažitelná jako typ vzhled hodinek, widget nebo jako typ aplikace. Aplikace ve všech režimech dokáže zobrazovat stejné metriky. Hlavním rozdílem je doba aktualizace dat, kdy u typu vzhledu hodinek probíhá každou minutu, u widgetu při každém aktivování obrazovky widgetu a u aplikace každou sekundu, tudíž se každá z těchto typů aplikací dá využít pro jiné účely. Zatímco aplikace typu vzhled hodinek a widget slouží pouze k zobrazení konfigurovaných metrik, samotný typ aplikace disponuje navíc možností volby profilů aktivit.

7.1 Vytvoření projektu

Při vývoji aplikace je nutné řešit zásadní problém s kompatibilitou, jelikož na rozdíl od mobilních telefonů, jsou zařízení Garmin velice variabilní. Je nutné se řídit dle toho, jakými senzory dané zařízení disponuje. Pokud zařízení disponuje například barometrem, je přístupná odpovídající třída, která senzor využívá, v opačném případě tato třída není dostupná. Existuje několik možností jak dané aplikace vyvíjet. V mé práci jsem zvolil vývojové prostředí Eclipse běžící na operačním systému Windows, jelikož zde tvůrci Monkey C připravili jednoduchý plugin pro vývoj aplikací. Pokud tedy máme nainstalováno Eclipse IDE, je jako první nutno doinstalovat XML editor s nástroji a poté provést samotnou instalaci zmíněného pluginu. Celý detailní postup instalace je možno nalézt v [23]. Největší předností tohoto pluginu je zvýraznění Monkey C syntaxe a poskytnutí simulátoru pro testování aplikací, přímo na námi vybraném zařízení. Abychom byli schopni vytvářet aplikace v Eclipse IDE, musíme dále stáhnout Connect IQ SDK a vygenerovat unikátní vývojářský klíč. Tento klíč je registrován ke konkrétnímu uživateli a není bez něho možné aplikace nahrávat do Connect IQ Store. Požadovaný klíč musí být privátní klíč RSA 4096 bit a je možné ho vygenerovat pomocí nástroje OpenSSL v kombinaci s příkazovým řádkem nebo přímo v pluginu Eclipse (klíč je následně uložen na disku).

Pro zhotovení Connect IQ aplikace vytvoříme v Eclipse IDE nový projekt, ve kterém je potřeba zvolit název projektu, umístění, typ aplikace, defaultní minimální SDK nutné pro běh aplikace, podporovaná zařízení a podporované jazykové mutace. Následně je vygenerován projekt, který obsahuje složky source, resources, bin a soubory manifest.xml (viz kapitola 4.1) a monkey.jungle (viz kapitola 4.2). Pro správné přeložení a spuštění aplikace je potřeba v Eclipse nastavit konfiguraci spuštění pro projekt Connect IQ. Při vytvoření nové konfigurace specifikujeme její název. Dále zvolíme projekt a aplikaci, která se s tímto nastavením bude spouštět, zařízení pro simulátor a verzi SDK. V takto nastaveném projektu můžeme vyvíjet naše aplikace.

7.2 Zvolené nástroje

K vytvoření konfigurovatelné aplikace jsem využil již zmíněné vývojové prostředí Eclipse s nainstalovaným XML editorem a pluginem Garmin se zabudovaným simulátorem pro

testování aplikací. Dále BFont [56] nebo-li Bitmap Font Generátor, pro importování vlastních stylů fontů. Pro účely testování jsem využil ještě dvojici aplikací Garmin Express a Garmin Connect Mobile.

Eclipse [55] je open source (počítačový software s otevřeným zdrojovým kódem) vývojová platforma, určená zejména pro programování v jazyce Java, nicméně pomocí pluginů lze rozšířit o programovací jazyky C++, PHP nebo právě Monkey C. Flexibilní návrh platformy Eclipse dále umožňuje také zápis HTML [57] (HyperText Markup Language) nebo XML kódu. Vývojové prostředí Eclipse ve své standardní verzi disponuje překladačem a debuggerem (nástroj pro hledání chyb). S nainstalovaným Garmin pluginem získává, také simulátor. V simulátoru je zobrazena testovaná aplikace, společně s informacemi o využití paměti aplikace. Dále je možné nastavit zobrazení času, použitý jazyk, simulovat připojení senzorů, spravovat ANT+ senzory, zaznamenávat vybranou aktivitu a především ovládat aplikaci stejně tak, jako bychom ji měli nainstalovanou na konkrétním fyzickém zařízení. Všechny data o aktivitě provedené na simulátoru je pak možné ukládat do souboru s příponou .FIT.

Bitmap Font Generátor (dále BFont) je open source program, který umožňuje generovat bitmapové fonty z TrueType fontů. TrueType je standard pro popis vektorových počítačových fontů, vyvinutý společnostmi Apple [58] a Microsoft [59] koncem 80.let. BFont umožňuje fonty dále upravovat, lze volit velikost mezer mezi znaky, obrys písma a další. Nejdůležitější funkcí je vygenerování fontu ve formátu TXT nebo PNG, jelikož právě tyto formáty fontů patří mezi podporované.

Garmin Express je počítačová aplikace, která je určena pro snadné nastavení, registraci a následnou správu zařízení Garmin. Po spárování zařízení s počítačem (USB kabelem nebo Bluetooth), se pomocí aplikace jednoduše synchronizují veškerá data z uložených aktivit na web, aktualizuje se firmware (software, který řídí systém hodinek) a dále je také možné aktualizovat aplikace nacházející se v našem zařízení.

Garmin Connect Mobile je mobilní aplikace, která je dostupná pro telefony s operačním systémem Android na Google Play [60] nebo operačním systémem iOS na App Store [61]. Lze zde najít informace o zdraví, historii počtu spálených kalorií, historii srdečního tepu, statistiky počtu kroků, detaily zaznamenaných aktivit a další. Po spárování se zařízením Garmin (skrze Bluetooth) je dále možné vytvářet tréninky, sestavovat trasy, sledovat ostatní uživatele, stahovat nové aplikace a nastavovat nainstalované aplikace. Garmin Connect Mobile je mobilní obdobou aplikace Garmin Express.

7.3 Definování problémů aplikace

Záměrem bylo především to, aby vytvořená aplikace fungovala a správně se zobrazovala na zařízeních jak s kulatým, tak hranatým displejem. Bylo však třeba zabývat se problematikou týkající se mrtvých rohů u kulatého displeje. Connect IQ SDK nijak neřeší tento problém a vývojáři aplikací mohou v nepozornosti vykreslovat komponenty i na místa, kde dané data nejsou vidět. Bylo tedy nutné vypočítat a přesně stanovit, kde vykreslit hraniční metriky, aby ještě zůstaly viditelné, zejména u zaoblených rohů displeje. Tento problém jsem vyřešil nastavením potřebného odsazení u hraničních pozic. Dalším problémem, který je s tímto úzce spjat, je velikost zobrazení dané metriky, jelikož má každá metrika jiný počet znaků a Monkey C neumožňuje dynamicky měnit velikost fontu v pixelech. Z tohoto důvodu bylo nutné vytvořit několik definovaných fontů s různou velikostí pro správné a hlavně nepřekreslující se zobrazení. Jeden typ fontu jsem využil na ikonky, aby při změně barevného schématu docházelo ke změnám barev také u nich. Dále bylo nutné řešit správné zpracování uživatelského vstupu při nastavování zobrazení daných metrik. Tento problém jsem vyřešil napsáním jednoduchého parseru, který čte pouze číselné znaky a pomlčku. V opačném případě by docházelo k pádu aplikace, jelikož neznalý uživatel může do pole v nastavení pozice zadávat jakékoli znaky. V neposlední řadě jsem musel omezit velikost používané paměti pro danou aplikaci, aby plynule fungovala. Tuto problematiku řeší programovací jazyk Monkey C, ve kterém se dá část kódu označit a následně rozdělit přímo pro konkrétní zařízení, jak je popsáno v kapitole 4.2. Detailní popis řešení zmíněných problémů uvádím dále v práci.

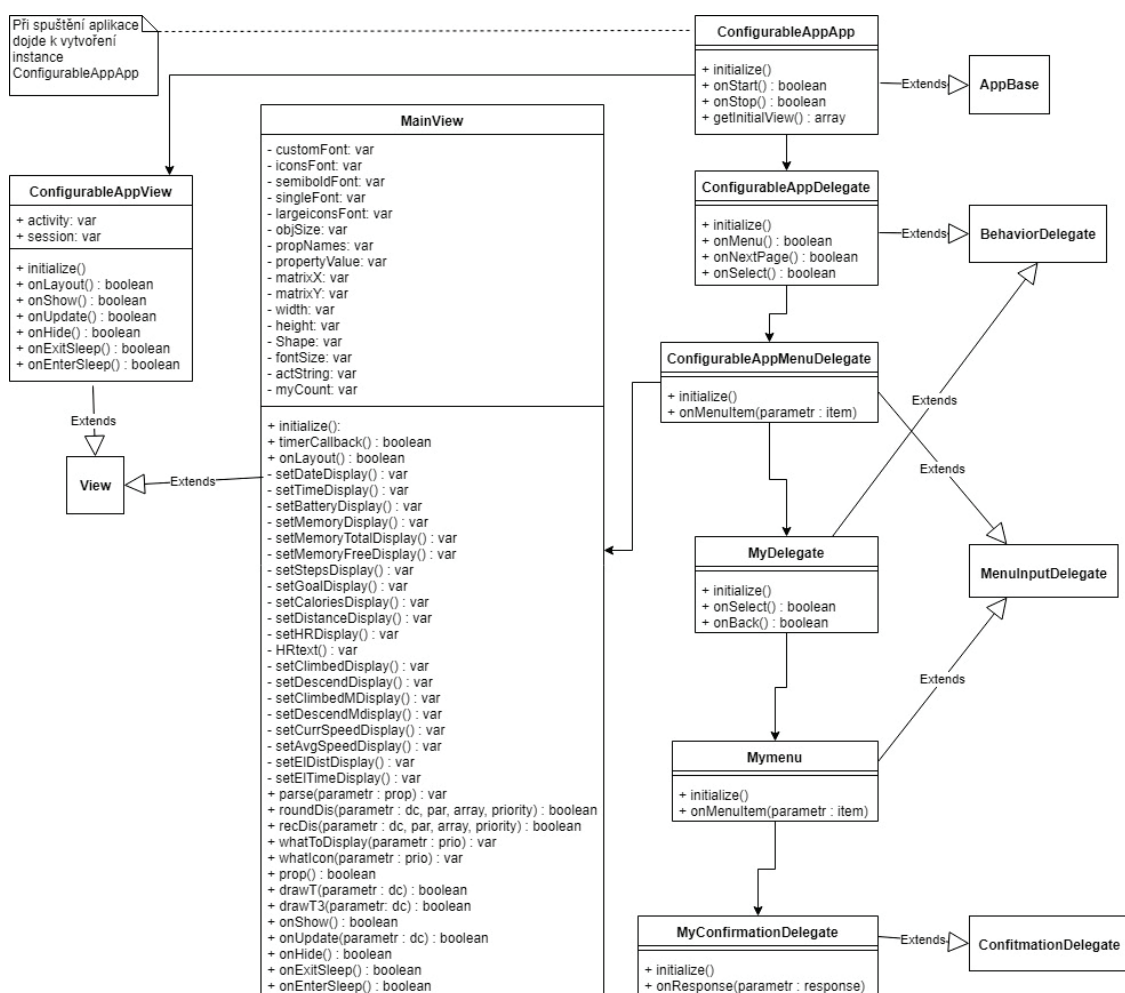
7.4 Popis implementace

Konfigurování aplikace probíhá ve zmíněných nástrojích Garmin Connect Mobile nebo Garmin Express. Veškeré konfigurované metriky jsou definovány, společně s nastavenými defaultními hodnotami ve zdrojovém adresáři `settings`, konkrétně v souborech `properties.xml` a `settings.xml` (viz kapitola 3.2). Aplikace je přeložena do dvou jazyků češtiny a angličtiny. Angličtinu jsem zvolil jako výchozí jazyk, ale v případě, že bude aplikace nahrána na zařízení s českou lokalizací, dojde k načtení zdrojů z adresáře `resources-ces`, která obsahuje překlady všech textů v aplikaci.

Ukázka na obrázku 1.15 znázorňuje třídy vyskytující se v mnou vytvořené konfigurovatelné aplikaci. U aplikací typu vzhled hodinek a widget by v tomto diagramu odpovídaly třídy `ConfigurableAppApp`, jenž je vyvolána při spuštění (zobrazení) aplikace, dále `ConfigurableAppDelegate` sloužící k zachycení uživatelské akce (stisknutí tlačítka) a `MainView`, kde probíhají hlavní výpočty hodnot z nastavených metrik a jejich následné zobrazení na displeji. Třída `MainView` u těchto dvou typů aplikací zároveň reprezentuje hlavní a jedinou obrazovku na displeji. V případě samotného typu aplikace je situace rozdílná, jelikož zde musí existovat třídy pro obsluhu jednotlivých nabídek menu a potvrzovacího dialogu.

Při spuštění aplikace dojde k vytvoření třídy `ConfigurableAppApp`, která rozšiřuje systémovou třídu `AppBase`. Tato třída pomocí metody `getInitialView()` vrátí počáteční

pohled obrazovky do třídy `ConfigurableAppView`, která rozšiřuje systémovou třídu `View` a delegáta, třídu `ConfigurableAppDelegate` rozšiřující třídu `BehaviorDelegate`. Následně je zobrazen vzhled úvodní obrazovky, pomocí metody `onLayout()` a ten zůstává aktivní, dokud není uživatelem stisknuto tlačítko další nebo start. Při detekci stisknutí tlačítka `ConfigurableAppDelegate` posune do popředí nabídkové menu (z adresáře zdrojů) s aktivitami a přesune obsluhu vstupu na novou třídu `ConfigurableAppMenuDelegate`, rozšiřující systémovou třídu `MenuInputDelegate`. Při výběru konkrétní položky menu dojde k nastavení globální proměnné `activity`, která uchovává číselnou hodnotu v závislosti na zvoleném typu aktivity (1 pro běh, 2 pro kolo a 3 pro turistiku). Po výběru aktivity je pohled přesunut do již zmíněné třídy `MainView`, která zařídí správné vypočítání a zobrazení metrik konfigurované uživatelem. Třídní delegát starající se o obsluhu vstupu v tomto pohledu je pojmenován `MyDelegate`.



Obrázek 1.15: Třídní diagram aplikace, vytvořený nástrojem draw.io [54]

Ve třídě `MainView` jsou definovány metody pro získání zobrazovaných metrik. Některá data jsou získávána z vestavěných senzorů (srdeční tep, akcelerometr a další), z toho důvodu bylo nutné do konfiguračního souboru `manifest.xml` přidat povolení, které informuje uživatele o této skutečnosti, viz také kapitola 4.1. Postup výpočtu k získání hodnoty například tepové frekvence vypadá následovně. První je zjištěno pomocí klíčového slova `INVALID_HR_SAMPLE` ze třídy `ActivityMonitor`, zda zařízení, na kterém je aplikace spuštěna disponuje senzorem snímače tepu. Pokud snímač není přítomen je v aplikaci reprezentován prázdným řetězcem, v opačném případě dochází k výpočtu. Při výpočtu je nutné zavolat metodu `getHeartRateHistory(period, newestFirst)`, která vrátí hodnotu tepové frekvence. Parametrem `period` nastaveným na hodnotu `null` (vrátí celou historii zaznamenaných hodnot tepové frekvence) a parametrem `newestFirst` nastaveným na hodnotu `true` (nejnovější vzorek tepové frekvence zobrazí jako první) je ze senzoru získána poslední zaznamenaná hodnota tepové frekvence. Takto vytvořeným metodám přiřazuji prioritu zobrazení (dle seřazení v nastavení aplikace) a případně grafickou ikonku, která se zobrazí vedle naměřené hodnoty. Grafické ikonky byly do aplikace nahrány skrze `BMFont`, tedy jako uživatelský font, který je vidět na obrázku 1.16.



Obrázek 1.16: *Náhled fontu ikonek, vygenerovaný programem BMFont*

V aplikaci jsou další tři mnou vytvořené fonty, které již obsahují standardní znaky a liší se od sebe pouze svou velikostí v pixelech. Správný font je použit v závislosti na orientaci zobrazení dané metriky (na výšku, na šířku nebo čtverec). Před zobrazením vypočítaných hodnot a ikonek je nutné zjistit, velikost nastavené matice, barevné provedení a rozmístění metrik zadáných uživatelem. Tyto údaje se zjišťují u všech třech typů mnou vytvořených konfigurovatelných aplikací s tím rozdílem, že u samotného typu aplikace, se pro každý zvolený profil aktivity získávají metriky samostatně. To ve výsledku znamená, že je uživatel schopný nakonfigurovat jinou velikost matice, barevné schéma a rozmístění metrik v této aplikaci pro každý profil aktivity zvlášť. Pro získání hodnoty zadané uživatelem v nastavení, slouží metoda `getProperty()` s jedním parametrem a to ID vyplněného pole, které je specifikováno v adresáři zdrojů (viz kapitola 3.2). Takto získané hodnoty především u rozmístění metrik je nutné dále zpracovat, jelikož nepozorný uživatel může do tohoto pole zadat nevalidní znaky. Z tohoto důvodu jsem v aplikaci vytvořil parser (syntaktický analyzátor), který kontroluje zadané hodnoty. Tento parser akceptuje pouze číselné znaky a za splněních podmínek také pomlčku (při detekci zadání intervalu). Jelikož nelze uživatele při konfigurování aplikace hned upozornit na chybně vyplněné pole (nečíselný znak nebo překrývající se senzory), tak toto řeším v rámci již spuštěné

aplikace a uživateli vypíše přes zobrazenou obrazovku informativní text s popisem chyby. Pokud se žádná chyba nedetekuje je ještě nutné před samotným vykreslením dat zjistit, jakým tvarem displeje disponuje zařízení, na kterém je aplikace spouštěna. Zavolá se metoda `getDeviceSettings()`, která vrátí informace o zařízení společně s hodnotou `screenShape` reprezentující tvar displeje. V závislosti na typu displeje se zavolá příslušná metoda k vykreslení metrik. Nemohla být použita pouze jednota komplexní metoda, jelikož v případě kulatého displeje, by vykreslovala metriky do rohů displeje, které nejsou viditelné. V tomto úseku je tedy zdrojový kód aplikace rozdělen do dvou větví. S využitím kvalifikátorů specifikovaných v hlavním `jungle` souboru, však nedochází při spuštění aplikace k překladu obou těchto částí, ale pouze odpovídajícímu úseku, který je určený právě tvarem displeje. U kulatého tvaru displeje je při vykreslování krajních hodnot nastaveno potřebné odsazení, aby nedošlo k vykreslení za hranu displeje. Pro samotné vykreslení hodnot je v závislosti na uživatelem zvolené velikosti matice (například 3x3, tedy 9 pozic) vypočítána pozice, dle velikosti displeje v pixelech. Pomocí těchto vypočítaných pozic, uložených v datovém typu `array`, se stanoví výsledná pozice zobrazovaných metrik.

Existuje několik typů jak se metriky mohou zobrazovat, dle zadaného čísla uživatelem u dané metriky, při konfiguraci aplikace. Uživatel může nastavit zobrazení metrik na jedné pozici (pouze jedno číslo) nebo přes více pozic, zadáním intervalu odděleného pomlčkou. V případě zadání intervalu se v aplikaci kontroluje, zda bude výsledná metrika roztažená na výšku, na šířku nebo bude mít čtvercový tvar. Pro vykreslení se použije metoda zmíněná v kapitole 4.4 `drawText()`, do které jsou předány vypočítané pozice metriky (parametry `x,y`), uživatelský font, zobrazovaná metrika (zvolená dle stanovené priority) a zarovnání. Aby nedocházelo k překreslení metrik na obrazovce hodinek je pozice vykreslené metriky obsazena (nastavením hodnoty na „x“). Stejný postup se opakuje, dokud na obrazovce nejsou vykresleny všechny metriky, které si uživatel nakonfiguroval v nastavení aplikace.

Pokud uživatel nacházející se na hlavní obrazovce stiskne tlačítko start vyvolá metodu `onSelect()`, která v závislosti na vybrané aktivitě (hodnota `activity`) spustí zaznamenávání aktivity. V aplikaci to znamená vytvoření objektu `session` pomocí metody `createSession()` a následné spuštění pomocí metody `session.start()`. V případě, že uživatel stiskne při zaznamenávání aktivity tlačítko zpět vyvolá se metoda `onBack()`, ve které se na objekt `session` zavolá metoda `session.lap()`, a tím je vytvořen nový úsek zaznamenávané aktivity, jak je popsáno výše v práci. Pokud uživatel zastaví zaznamenávání aktivity opětovným stisknutím tlačítka start, pak je vyvolána metoda `session.stop()` a pohled obrazovky je přesunut do třídy `MyMenu`. V této třídě je uživateli na obrazovce zobrazeno menu s třemi položkami, jmenovitě pokračovat, uložit a zahodit. Při zvolení tlačítka pokračovat je pouze zavolána metoda `session.start()` a pohled je navrácen zpět na zaznamenanou aktivitu. Pokud uživatel zvolí možnost uložení je zavolána metoda `session.save()`, následně je objekt `session` nastaven na hodnotu `null` a pohled navrácen na obrazovku vyresetované aktivity. V posledním případě je obrazovka uživatele přesunuta do třídy `MyConfirmationDelegate` rozšiřující třídu `ConfirmationDelegate`, která vyvolá

potvrzovací dialog s možnostmi potvrdit nebo zrušit. Při potvrzení dojde v aplikaci k vyvolání metody `session.discard()`, objekt `session` je nastavený na hodnotu `null` a pohled navrácen do menu aktivit. V opačném případě je uživatel vrácen na původní nabídku.

8 Testování

Testování vytvořených aplikací probíhalo odděleně ve dvou skupinách. V první skupině byly zvlášť testovány aplikace typu vzhled hodinek a widget, ve druhé skupině samotný typ aplikace. Testování vytvořených prototypů aplikací probíhalo v simulátoru vývojového prostředí Eclipse.

Zobrazení bylo testováno na zařízeních s dvěma rozdílnými velikostmi displejů. Na kulatém displeji s velikostí 215 x 180px, který podporuje zařízení fr230, fr235, fr735XT a foreathlete735XTJ. Dále pak zařízení s hranatým displejem s velikostí 145 x 205px vivoActive HR. Jelikož patří testované rozměry kulatého displeje mezi nejmenší Garmin zařízení, předpokládá se, že aplikace bude správně zobrazována i na zařízeních typu Fenix 3, Fenix 5, Fenix 5S a dalších, které také disponují kulatým displejem se stejným poměrem stran, ale s většími rozměry. Při testování typu aplikace jsem navíc musel ošetřit zobrazení textu na úvodní obrazovce, aby nedošlo k vykreslení na slepá místa, která jsou za hranou displeje.

Funkcionalitu konfigurovatelné aplikace jsem testoval postupným nastavováním rozdílného rozložení metrik, zadáváním nevalidních intervalů, zadáváním nečíselných znaků a další. Při testování typu aplikace jsem musel validovat jednotlivé nabídky menu (menu s výběrem aktivit a menu při vypnutí zaznamenávání aktivity) a potvrzovací dialog. Zejména bylo nutné posunout správný pohled, po interakci s nabídkou menu na obrazovku v popředí.

Při testování jsem na simulátoru postupně vyřešil všechny problémy zmíněné v kapitole 7.2 a dalším krokem bylo otestování aplikací již na konkrétním fyzickém zařízení. Všechny typy aplikací jsem postupně umístil na Garmin Store IQ, kde v několika dnech prošly schvalovacím procesem a nyní jsou volně ke stažení na odkazech [26] jako vzhled hodinek, [27] jako widget a [45] jako typ aplikace. Hned po prvním stažení a následném nahrání konfigurovatelné aplikace do mého zařízení fr235 jsem zjistil, že defaultní font, který jsem zvolil je kvůli své malé velikosti špatně čitelný. Z tohoto důvodu jsem byl nucen vytvořit a nahrát zcela novou sadu fontů s třemi velikostmi pro lepší čitelnost. Dále jsem musel opravit pozicování grafických ikon, jelikož se na zařízení zobrazovaly na jiném místě než v simulátoru. Obecně jsem zjistil, že výsledný vzhled reálné aplikace se liší od vzhledu nabízeného v Eclipse simulátoru. Velikost obrazovky v simulátoru přesně pixelově neodpovídá zobrazení, jako na reálném zařízení a může tedy dojít k odchylkám v tomto zobrazení. Pro úplnou korektnost jsem tedy musel veškeré grafické prvky testovat přímo na konkrétním fyzickém zařízení. Konfigurace aplikací probíhala vždy za pomoci zmíněných aplikací Garmin Express nebo Garmin Connect Mobile. Při každé úpravě konfigurace bylo nutné zařízení znovu synchronizovat, aby se změněné metriky projevíly v aplikaci.

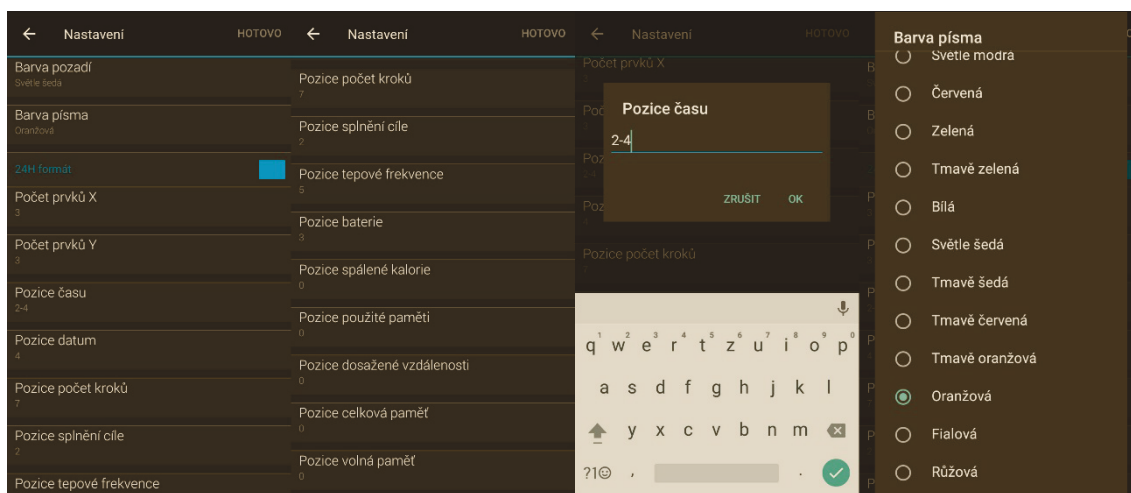
8.1 Směr vývoje

Výsledné aplikace splňují požadavky zadání ve všech bodech i přesto by do budoucna bylo vhodné s vývojem konfigurovatelné aplikace nadále pokračovat. Mým dalším návrhem pro budoucí vývoj, by v první řadě bylo rozšíření aplikací pro všechny typy dostupných zařízení Garmin a přidání dalších jazykových mutací. Tyto kroky by vedly k tomu, že bude aplikace

dostupná většímu počtu potenciálních uživatelů. Dále bych se zaměřil na vytvoření přívětivější grafické stránky aplikace, aby nové uživatele zaujala na první pohled nebo také možnost volit z více fontů zobrazení. Pomocí jednoduché mobilní aplikace, by se dalo docílit i možnosti nahrávání uživatelských fontů do mnou vytvořené aplikace a tak docílit ještě větší variability. Dalším přínosem pro uživatele, by byl mobilní telefon připojený k internetu v dosahu Bluetooth signálu Garmin zařízení, po celou dobu běhu aplikace. Otevřelo by to novou cestu k datům a informacím, jenž by konfigurovatelná aplikace mohla zobrazovat.

8.2 Ukázky a srovnání aplikace

Ukázka na obrázku 1.17 znázorňuje nastavení v mobilní aplikaci Garmin Connect Mobile, pro aplikace typu vzhled hodinek nebo widget. Na první a druhé obrazovce tohoto obrázku, lze vidět metriky, které si uživatel může nakonfigurovat na zařízení fr235. Na třetí obrazovce tohoto obrázku je konkrétní konfigurace pozice a velikosti zobrazení metriky času, zadané intervalem. Tato konfigurace vypadá u všech ostatních metrik stejně. Na poslední obrazovce je zobrazená nabídka s volbou barevného provedení fontu.



Obrázek 1.17: Obrazovky znázorňující nastavení v mobilní aplikaci Garmin Connect Mobile

Na obrázku 1.18 vidíme ukázku již nakonfigurované aplikace s následujícími hodnotami: velikost matice 3x3, pozice 1 vzdálenost, pozice 2-3 rychlost, pozice 4 stav baterie, pozice 5 srdeční tep, pozice 6 spálené kalorie a pozice 7-9 čas. Tato ukázka odpovídá zobrazení již zaznamenávané aktivity.



Obrázek 1.18: Ukázka zobrazení dat v aplikaci, obrázek pořízen ze simulátoru

Nyní budu srovnávat postupně všechny vytvořené typy aplikací s aplikacemi, které jsou nejvíce podobné. Porovnávám aplikace jsem zvolil na základě funkcionality, oblíbenosti a počtu stažení na Garmin Store IQ.

Pokud srovnám mnou vytvořenou konfigurovatelnou aplikaci typu vzhledu hodinek se zmíněnou nejstahovanější aplikací NoFrills [53], docházím k závěru, že má tato aplikace propracovanější grafický vzhled a provedení (viz obrázek 1.19). Tato aplikace podporuje téměř veškerá Garmin zařízení a je přeložena do 20 jazyků. Pokud se však podívám na stránku konfigurovatelnosti zobrazení jednotlivých metrik a celkové nastavení aplikace, má mnou vytvořená aplikace výhodu v tom, že lze konfigurovat všechny zobrazované metriky, zatímco u zmíněné aplikace NoFrills se dají měnit pouze barvy, případně zde lze některé údaje skrýt. Výsledný vzhled této aplikace bude však vždy totožný s minimálními rozdíly. Na přiloženém obrázku 1.19, lze vidět zobrazení vybraných metrik. Zatímco v aplikaci NoFrills bude vždy čas a datum zobrazen na stejném místě, v mé aplikaci je možné pozici a velikost těchto metrik zvolit v nastavení.



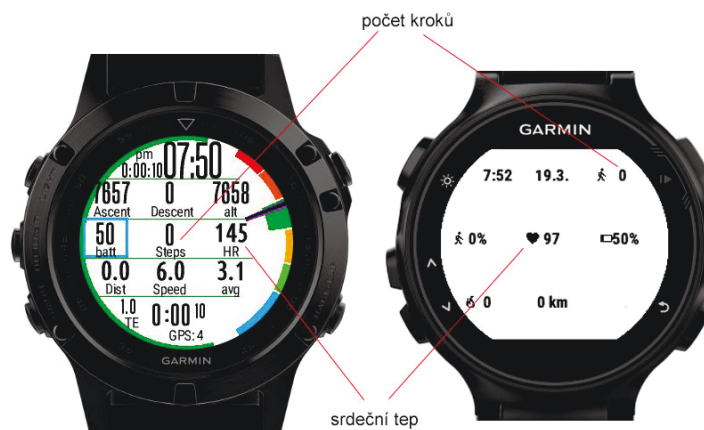
Obrázek 1.19: Porovnání aplikace NoFrills s konfigurovatelnou aplikací typu vzhled hodinek

Při hledání podobné aplikace typu widget jsem na Garmin Store IQ našel aplikaci s názvem Weekly Activity [62]. Tato aplikace má ve svém popisu, že zobrazuje týdenní statistiky jako počet kroků, vzdálenost, kalorie a další. Byla tedy vhodným kandidátem ke srovnání s mnou vytvořeným konfigurovatelným widgetem. Zjistil jsem, že aplikace Weekly Activity podobně jako aplikace NoFrills, podporuje většinu Garmin zařízení a je přeložena do 10 jazyků. Další výhodou je, že aplikace je na Garmin Store IQ publikována již od září roku 2017 a je neustále vyvíjena, což dokládá i poslední aktualizace z dubna roku 2019. Pokud se podíváme na nastavení této aplikace je velmi detailní i ve své neplacené verzi a můžeme si zde zvolit, které metriky chceme zaznamenávat. Ukázku lze vidět na obrázku 1.20, kde je v aplikaci Weekly Activity zobrazen týdenní přehled dosažené vzdálenosti v kilometrech. Tento přehled je doplněn celkovou sumou (symbol Σ) dosažených kilometrů za posledních sedm dní, znázorněnou na spodní straně obrazovky. Zároveň je možné tento přehled zobrazit pro další podporované metriky widgetu zmíněné v úvodu. Samotné zobrazení této aplikace však zůstává stejné a měnit můžeme pouze barevné provedení. Hlavní výhodou mého widgetu tedy stejně jako u typu aplikace vzhled hodinek zůstává konfigurovatelnost zobrazení daných metrik.



Obrázek 1.20: Porovnání aplikace *Weekly Activity* s konfigurovatelnou aplikací typu *widget*

Posledním typem mnou vytvořené aplikace je samotný typ aplikace, kterou budu srovnávat s multisportovní aplikací Sports Monitor [63]. Tato aplikace slouží k zaznamenávání sportovní aktivity, nezáleží tedy na konkrétním druhu prováděné aktivity, zatímco v mé aplikaci si uživatel v menu musí zvolit ze třech nabízených aktivit (běh, kolo nebo turistika). Sports Monitor má konfigurovatelné metriky: zóny srdečního tepu, rychlost, tempo a zvuky (rozdílné tóny). Dále tato aplikace umožňuje definovat metrický formát (US míle nebo EU km). Stejně jako mnou vytvořená konfigurovatelná aplikace, podporuje pouze dva jazyky. V těchto ohledech jsou si tedy tyto dvě aplikace velmi podobné. Nevýhodou Sports Monitor je, že podporuje pouze zařízení s identickou velikostí a tvarem displeje jako má model Fenix 5. Vizuální srovnání lze vidět na obrázku 1.21, kde je zobrazena obrazovka již spuštěné aktivity. Zde bych zmínil především metriku znázorňující srdeční tep, jelikož je zobrazena jak číselnou hodnotou, tak grafickým panelem na pravém okraji hodinek, který zároveň ukazuje zóny srdečního tepu (zóna udává zátěž). Všechny ostatní metriky aplikace Sports Monitor zobrazené na obrázku 1.21 mají přesně definovanou pozici a velikost zobrazení, které nelze měnit. Toto je hlavním rozdílem od mé aplikace, která podporuje stejné metriky, ale zároveň dává uživateli možnost konfigurace pozice a velikosti.



Obrázek 1.21: Porovnání aplikace Sports Monitor s konfigurovatelnou aplikací

Při porovnávání mé konfigurovatelné aplikace s již existujícími jsem došel vždy ke stejnému závěru a to, že všechny srovnávané aplikace neposkytují uživateli konfigurovatelnost zobrazovaných metrik. U těchto zobrazovaných metrik nelze měnit velikost ani pozici a výsledný vzhled aplikací zůstává, až na grafické detaily stejný. Aplikace uživateli nabízí pouze možnost danou metriku nezobrazovat. Pozitiva v porovnávaných aplikacích shledávám v tom, že disponují propracovaným grafickým vzhledem a jelikož jsou vyvíjeny desítky měsíců mají také velký počet stažení a zpětné vazby od uživatelů. Aplikace jsou stabilní a nijak nezpomalují zařízení, na kterém jsou nainstalovány. Pokud bych v mnou vytvořené konfigurovatelné aplikaci zapracoval dále na přívětivějším grafickém vzhledu v podobě, kde by si uživatel zvolil rozlišnou barvu pro každou ze zobrazených metrik. A dále bych graficky upravil rozdělení daných metrik přidáním mřížky, která by definovala velikost jejich zobrazení a přeložil aplikaci do více jazyků, mohla by konkurovat porovnávaným aplikacím ve všech ohledech.

Závěr

Cílem této bakalářské práce bylo nastudovat platformu Garmin Connect IQ, navrhnout komponenty uživatelského rozhraní zobrazující metriky ze senzorů zařízení a následně vytvořit konfigurovatelnou aplikaci hodinek tří typů a to jako vzhled hodinek, widget a typ aplikaci.

V úvodní části této práce je popsán historický vývoj zařízení od společnosti Garmin a srovnání dvou řad jmenovitě Forerunner, určenou zejména pro běžce, a multisportovní vívoActive. Pro testování aplikace byl tento výběr přínosný, jelikož ukázal problémy v rozdílném zobrazení u daných zařízení. Dále byl popsán programovací jazyk Monkey C, který je určen k vývoji aplikací na zařízeních Garmin. Zjistil jsem, že je tento jazyk velice podobný jazykům, se kterými jsem se dosud setkal, zejména bych jej přirovnal k jazyku C++. Nebylo tedy potřeba učit se zcela nový jazyk, ale spíše dohledat menší rozdíly mezi těmito jazyky. V neposlední řadě jsem se seznámil s podporovanými senzory zařízení a jejich algoritmy pro získávání hodnot.

V praktické části mé práce je popsáno samotné řešení všech typů mnou vytvořené konfigurovatelné aplikace pro hodinky Garmin Forerunner 235 a Garmin vívoActive HR. Byly zde analyzovány jednotlivé problémy a jejich řešení, popsány zvolené nástroje, samotná implementace, testování a porovnání s podobnými existujícími aplikacemi. Zjistil jsem, že pro úplné využití konfigurovatelné aplikace je potřeba využít programy Garmin Express na počítačích nebo Garmin Connect Mobile na mobilních telefonech, jelikož samotná zařízení nemají dostatečnou paměť pro uchovávání konfigurace nastavení. Skrze tyto programy se uživatel dostane k nastavení metrik mnou vytvořené aplikace a pouze tyto nastavené metriky jsou synchronizovány do daného zařízení.

Hlavním výsledkem práce je konfigurovatelná aplikace, která je stažitelná jako vzhled hodinek, widget nebo typ aplikace. Pokud tedy uživatel cílí na vysokou výdrž baterie, ale zároveň má zájem data stále vidět, využije typ aplikace vzhled hodinek, jelikož zůstane na stejné obrazovce a zobrazované metriky se aktualizují každou minutu. Zatímco typ widget využijí zejména ti uživatelé, kteří chtějí údaje ze senzorů zobrazovat až při dotázání, nebo-li přejetí na obrazovku s widgetem. Samotný typ aplikace je po zapnutí podobný vzhledu hodinek, ale aktualizace zde probíhají každou sekundu. Použití je tedy velice flexibilní a zásluhou uživatelských vstupů také velice variabilní. Všechny typy aplikací byly nahrány na oficiální Garmin Store IQ, kde byly ve zkušebním provozu několik týdnů a desítky uživatelů je během této doby stáhlo do svého zařízení. Na základě ohlasů a kontroly výstupů, byla prováděna nejrůznější vylepšení a aplikace se stala uživatelsky přívětivější. Během zkušebního provozu se nevyskytl žádný závažný problém. Při srovnávání s již vytvořenými konfigurovatelnými aplikacemi ostatních vývojářů jsem zjistil, že jsou tyto aplikace vytvářeny přesně na konkrétní Garmin zařízení a neumožňují uživateli konfigurovat všechny zobrazované metriky. Tyto aplikace jsou navíc vytvořeny jako jeden konkrétní typ aplikace a tím ztrácí svou další využitelnost, na rozdíl od mnou vytvořené konfigurovatelné aplikace. Tvorba a realizace této práce pro mě představovala velice cennou zkušenost v oblasti návrhu a vývoje aplikací.

Použitá literatura

- [1] Polar history [online]. 2017, poslední revize 01.04.2019 [cit. 2019-01-02]. Dostupné z: <https://www.polar.com/blog/40-years-of-incredible-firsts-polar-history/>
- [2] Suunto [online]. c2019 [cit. 2019-01-02]. Dostupné z: <https://www.suunto.com/cs-cz/>
- [3] ISO 639-2 Language Code List [online]. 2013, poslední revize 21.12.2017 [cit. 2019-01-01]. Dostupné z: https://www.loc.gov/standards/iso639-2/php/code_list.php
- [4] Garmin fēnix 5 [online]. c1996-2019 [cit. 2019-01-02]. Dostupné z: <https://buy.garmin.com/en-US/US/p/552982#overview>
- [5] Garmin products [online]. 2011, poslední revize 10.02.2011 [cit. 2019-01-02]. Dostupné z: <http://www.path.cz/forum/viewtopic.php?f=33&t=1232>
- [6] Garmin actual products [online]. c1996-2019 [cit. 2019-01-02]. Dostupné z: <https://www.garmin.com/en-US/>
- [7] Garmin quarterly and annual earnings [online]. c1996-2019 [cit. 2019-01-02]. Dostupné z: <https://www.garmin.com/en-US/company/investors/earnings/>
- [8] Garmin Forerunner 235. Garmin [online]. c1996-2019 [cit. 2019-01-02]. Dostupné z: <https://buy.garmin.com/en-US/US/p/529988#overview>
- [9] Garmin Vivoactive HR. Garmin [online]. c1996-2019 [cit. 2019-01-02]. Dostupné z: <https://buy.garmin.com/en-US/US/p/pn/010-01605-03>
- [10] Wear OS by Google [online]. c2019 [cit. 2019-01-06]. Dostupné z: <https://developer.android.com/wear/>
- [11] Movesense iOS and Android Libraries [online]. 2017, poslední revize 1.10.2018 [cit. 2019-01-06]. Dostupné z: <https://bitbucket.org/suunto/movesense-mobile-lib>
- [12] Connect IQ SDK. Garmin [online]. c2019 [cit. 2019-01-06]. Dostupné z: <http://developer.garmin.com/connect-iq>
- [13] Monkey C. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/monkey-c/>
- [14] Resource compiler. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/programmers-guide/resource-compiler/>
- [15] User Experience Guide. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/user-experience-guide/>
- [16] API Docs. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/api-docs/>

-
- [17] Manifest file. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/programmers-guide/entry-points/>
- [18] Overriding Resources. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <https://developer.garmin.com/connect-iq/programmers-guide/overriding-resources/>
- [19] Garmin Connect API. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <http://developer.garmin.com/garmin-connect-api/overview/>
- [20] User interface. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <http://developer.garmin.com/connect-iq/programmers-guide/user-interface/>
- [21] Positioning and sensors. Garmin [online]. c2019 [cit. 2019-01-10]. Dostupné z: <http://developer.garmin.com/connect-iq/programmers-guide/positioning-sensors/>
- [22] Wear Where Ware, Connect IQ and the wearable world. Garmin [online]. c2019 [cit. 2019-02-18]. Dostupné z: <https://developer.garmin.com/downloads/connect-iq/Wear%20Where%20Ware%20-%20Connect%20IQ%20and%20the%20wearable%20world.pdf>
- [23] Getting started. Garmin [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://developer.garmin.com/connect-iq/programmers-guide/getting-started/>
- [24] Garmin Connect. Garmin [online]. c1996-2019 [cit. 2019-03-20]. Dostupné z: <https://connect.garmin.com/en-US/>
- [25] Garmin Express. Garmin [online]. c1996-2019 [cit. 2019-03-20]. Dostupné z: <https://www.garmin.com/cs-CZ/software/express>
- [26] Configurable Watch Face. Garmin [online]. c1996-2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/en-US/apps/0b239f69-ca9d-458b-87eb-8103ecad41d1>
- [27] Configurable Widget. Garmin [online]. c1996-2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/en-US/apps/905cc4e7-7988-4578-a58b-c71c378bd9c8>
- [28] C programming language [online]. c2019 [cit. 2019-04-03]. Dostupné z: https://www.tutorialspoint.com/c_standard_library/index.htm
- [29] Java programming language [online]. c1993-2018 [cit. 2019-04-03]. Dostupné z: <https://docs.oracle.com/javase/7/docs/technotes/guides/language/>
- [30] Javascript scripting language [online]. c2019 [cit. 2019-04-03]. Dostupné z: <https://www.w3schools.com/js/>
- [31] Python programming language [online]. c2001-2019 [cit. 2019-04-03]. Dostupné z: <https://www.python.org/>
- [32] Lua programming language [online]. c2019 [cit. 2019-04-03]. Dostupné z: <https://www.lua.org/>

-
- [33] Ruby programming language [online]. c2019 [cit. 2019-04-03]. Dostupné z: <https://www.ruby-lang.org/en/>
- [34] PHP scripting language [online]. c2001-2019 [cit. 2019-04-03]. Dostupné z: <https://www.php.net/>
- [35] ANT+ device profiles [online]. c2019 [cit. 2019-04-04]. Dostupné z: <https://www.thisisant.com/developer/ant-plus/device-profiles>
- [36] Movesense senzor datasheet [online]. c2019, poslední revize 9.11.2017 [cit. 2019-04-04]. Dostupné z: <https://www.movesense.com/wp-content/uploads/2017/11/Movesense-Sensor-Datasheet--20171109.pdf>
- [37] Di2 integration [online]. 2018, poslední revize 14.2.2018 [cit. 2019-04-04]. Dostupné z: <https://bike.shimano.com/en-US/information/news/di2-integration--the-world-s-smartest-bicycle.html>
- [38] Garbage Collection vs Automatic Reference Counting [online]. c2018, poslední revize 1.2.2018 [cit. 2019-04-04]. Dostupné z: <https://medium.com/computed-comparisons/garbage-collection-vs-automatic-reference-counting-a420bd4c7c81>
- [39] Duck typing [online]. 2017, poslední revize 28.8.2018 [cit. 2019-04-04]. Dostupné z: <https://devopedia.org/duck-typing>
- [40] Strava Relative Effort application [online]. c1996-2019 [cit. 2019-04-04]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/6b53eedd-bf67-4c18-a2d6-af1d59518357>
- [41] dwMap application [online]. c1996-2019 [cit. 2019-04-04]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/2750f280-82f4-4f21-a32c-57acc7ce4870>
- [42] Garmin vivoActive 3 Music [online]. c1996-2019 [cit. 2019-04-04]. Dostupné z: <https://www.garmin.cz/garmin-vivoactive3-music-optic-black-band/79581>
- [43] Garmin Forerunner 645 Music [online]. c1996-2019 [cit. 2019-04-04]. Dostupné z: <https://buy.garmin.com/en-US/US/p/612476>
- [44] Spotify application [online]. c1996-2019 [cit. 2019-04-04]. Dostupné z: <https://apps.garmin.com/en-US/apps/30c6c876-ba43-4cbb-b4c7-03583a7cb66b>
- [45] Configurable application. Garmin [online]. c1996-2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/aa6685d7-6dec-4087-bc8c-c72336ade84e>
- [46] Bluetooth technology. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.bluetooth.com/>
- [47] Wi-Fi technology. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.wi-fi.org/>
- [48] Polar earnings. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.crunchbase.com/organization/polar-electro#section-competitors-revenue-by-owler>
-

-
- [49] Suunto earnings. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.crunchbase.com/organization/suunto#section-competitors-revenue-by-owler>
- [50] Wear OS improving battery life. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://support.google.com/wearos/answer/6303917?hl=en>
- [51] Battery drain on Wear OS. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://support.google.com/wearos/thread/590375?hl=en>
- [52] Build with Connect IQ. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://developer.garmin.com/connect-iq/what-you-can-build/>
- [53] NoFrills watchface application. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/03030574-3c6e-484a-9bd8-ce2ca0249651>
- [54] Draw.io online diagram creator. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/03030574-3c6e-484a-9bd8-ce2ca0249651>
- [55] Eclipse IDE. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.eclipse.org/ide/>
- [56] Bitmap Font Generator. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.eclipse.org/ide/>
- [57] HTML tutorial. [online]. c1999-2019 [cit. 2019-03-20]. Dostupné z: <https://www.w3schools.com/html/>
- [58] Apple corporation. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.apple.com/>
- [59] Microsoft corporation. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.microsoft.com/>
- [60] Garmin Connect Mobile on Google Play. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.garmin.android.apps.connectmobile&hl=cs>
- [61] Garmin Connect Mobile on App Store. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://itunes.apple.com/us/app/garmin-connect/id583446403?mt=8>
- [62] Weekly Activity widget. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/222eb4ce-8387-49c3-9560-1334ce1aeb4>
- [63] Sports Monitor application. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/33732e76-1f6a-4bed-a4c1-65da5df2a0a2>
- [64] iOS. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.apple.com/ios/ios-12/>
- [65] Movesense platform. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.movesense.com/>

-
- [66] C++ programming language. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <http://www.cplusplus.com/>
- [67] Garmin Varia Vision. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://www.garmin.cz/garmin-varia-vision---head-up-displej/78233>
- [68] Sun & Moon widget. [online]. c2019 [cit. 2019-03-20]. Dostupné z: <https://apps.garmin.com/cs-CZ/apps/91c09c18-b573-4d9a-81ac-df4bbe2298dc>
- [69] Military Grid Reference System. [online]. c1999-2018 [cit. 2019-04-20]. Dostupné z: https://www.armystudyguide.com/content/Prep_For_Basic_Training/Prep_for_basic_land_navigation/locate-a-point-using-the-.shtml
- [70] Latitude and longitude coordinates. [online]. c2019 [cit. 2019-04-20]. Dostupné z: <https://gisgeography.com/latitude-longitude-coordinates/>
- [71] Apple watch battery life. [online]. c2019 [cit. 2019-04-28]. Dostupné z: <https://www.apple.com/watch/battery.html>
- [72] Pebble battery life. [online]. 2019, poslední revize 04.03.2016 [cit. 2019-04-28]. Dostupné z: <http://pebble-help-legacy.rebble.io/help.getpebble.com/customer/portal/articles/1564016-battery-life.html>
- [73] Approach® S10. [online]. c1996-2019 [cit. 2019-04-28]. Dostupné z: <https://buy.garmin.com/en-US/US/p/611037#specs>